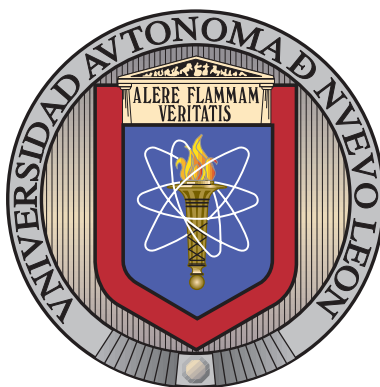


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



UNA HEURÍSTICA BASADA EN BÚSQUEDA
ESTOCÁSTICA PARA PROBLEMAS DE
OPTIMIZACIÓN GLOBAL

POR

JONÁS VELASCO ÁLVAREZ

EN OPCIÓN AL GRADO DE

MAESTRO EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

MARZO DE 2009

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



UNA HEURÍSTICA BASADA EN BÚSQUEDA
ESTOCÁSTICA PARA PROBLEMAS DE
OPTIMIZACIÓN GLOBAL

POR

JONÁS VELASCO ÁLVAREZ

EN OPCIÓN AL GRADO DE

MAESTRO EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

MARZO DE 2009

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis «Una heurística basada en búsqueda estocástica para problemas de optimización global», realizada por el alumno Jonás Velasco Álvarez, con número de matrícula 1437957, sea aceptada para su defensa como opción al grado de Maestro en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis

Dr. J. Arturo Berrones Santos

Asesor

Dra. Satu Elisa Schaeffer

Revisor

Dr. Roger Z. Ríos Mercado

Revisor

Vo. Bo.

Dr. Moisés Hinojosa Rivera

Subdirector

División de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, marzo de 2009

*Este trabajo se lo dedico a las personas que amo y que siempre han creído en mí,
me han dado su apoyo incondicional y que son mi motivo para seguir adelante.*

A mis padres,

*El Sr. René Velasco Flores y
la Sra. Rosa Martha Alvarez López.*

A mis hermanos,

*Noé Velasco Alvarez,
René Velasco Alvarez y
Martha Elena Velasco Alvarez.*

y a mi hija,

Azul Zoé Velasco Orozco.

ÍNDICE GENERAL

Agradecimientos	XIII
Resumen	xv
1. Introducción	1
1.1. Objetivo	1
1.2. Justificación	1
1.3. Hipótesis	2
1.4. Estructura de la tesis	2
2. Antecedentes	4
2.1. Algoritmos de búsqueda estocástica en la optimización global	5
3. Estimación de densidades de probabilidad	7
3.1. Densidades de probabilidad en un proceso de búsqueda estocástica	7
3.1.1. El SFPL	9
3.1.2. El algoritmo SFPL	13
3.1.3. Parámetros del algoritmo SFPL	13

3.2. El SFPL para problemas sin restricciones	15
3.2.1. Densidades de las funciones	15
3.2.2. Convergencia de las densidades del SFPL	16
3.3. Conclusiones del capítulo	21
4. Estudio del SFPL en problemas con restricciones	22
4.1. Antecedentes	22
4.2. Método de penalización	22
4.2.1. El problema de la mochila	23
4.3. SFPL: un enfoque bicriterio	29
4.4. SFPL: una densidad conjunta	31
4.5. SFPL: Densidad para la función de restricciones	34
4.6. Los Algoritmos Genéticos	38
4.6.1. Antecedentes	38
4.6.2. Generalidades	40
4.7. Hibridación de SFPL con Algoritmos Genéticos	43
4.7.1. Población	44
4.7.2. Función Objetivo	45
4.7.3. Selección	46
4.7.4. Cruce	47
4.7.5. Mutación	48
4.7.6. Parámetros	48

4.7.7. Algoritmo de SFPL-AGI	49
4.7.8. Algoritmo de SFPL-AGII	51
4.8. Un enfoque de manejo de restricciones en Algoritmos Genéticos . . .	54
4.9. Conclusiones del capítulo	56
5. Experimentación computacional	58
5.1. Análisis estadístico	59
5.1.1. Función de Booth	59
5.1.2. Función de Rosenbrock	61
5.1.3. Función de Schwefel	63
5.2. Costo computacional y dimensionalidad	65
5.2.1. Algoritmos genéticos	65
5.2.2. SFPL híbrido	66
5.3. Conclusiones del capítulo	69
6. Conclusiones	70
6.1. Contribuciones	70
6.2. Conclusiones	71
6.3. Trabajo futuro	71
A. Método del Gradiente	74
B. Recocido Simulado	77
C. Lista de términos y definiciones	79

ÍNDICE DE FIGURAS

3.1. N partículas que interactúan en un potencial	8
3.2. Densidad de una partícula en un potencial.	9
3.3. Estimación de las densidades de x_1 y x_2 para la función de Booth, usando parámetros de $L = 100$ y $D = 10$	16
3.4. Convergencia de las densidades a una correcta distribución al estado estacionario en la función de Booth para x_1 y x_2 con 5, 50 y 100 iteraciones del SFPL. Se utilizó los parámetros de $L = 100$ y $D = 10$	18
3.5. Se ilustra la convergencia de las densidades estimadas por el SFPL para el problema del vendedor de periódico. Las iteraciones del SFPL para x son de 2, 5, 20, 50, 100, 200, 500, 800 de izquierda a derecha. Se utilizaron los parámetros $L = 50$ y $D = 10$	20
4.1. Estimación de las densidades de probabilidad para el problema de la mochila de tres variables con parámetros para el SFPL de $L = 100$ y $D = 1$	26
4.2. Estimación de las densidades de probabilidad para el problema de la mochila de diez variables con parámetros para el SFPL de $L = 100$ y $D = 0.5$	28

4.3. Estimación de las densidades asociadas a la función objetivo de costos $p(x_n f_c)$ para el problema de prueba P.1 , con parámetros de $L = 100$ y $D = 0.5$ para el SFPL.	33
4.4. Estimación de las densidades asociadas a la función de restricciones $p(x_n f_r)$ para el problema de prueba P.1 , con parámetros $L = 200$ y $D = 0.03$ para el SFPL.	33
4.5. La densidad conjunta $p(x_n f_c)p(x_n f_r)$ obtenida para el problema de prueba P.1 . La región de alta probabilidad de las densidades corresponden al óptimo global conocido.	34
4.6. Diagrama general de un algoritmo genético.	39
4.7. Operador de cruce para el algoritmo genético basado en un punto de corte.	41
4.8. Densidad estimada por el SFPL que es muestreada en la región de alta probabilidad	45
4.9. Operador de cruce aritmético para el algoritmo genético.	48
4.10. Comparación de la convergencia del AGS contra el algoritmo SFPL-AGI para el problema de Booth.	51
4.11. Convergencia del algoritmo SFPL-AGI según su valor objetivo para el problema de prueba P.1.	55
5.1. Función de Booth en 2D, $f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$. . .	60
5.2. Función de Rosenbrock en 2D, $f(x, y) = 100(y - x^2)^2 + (x - 1)^2$. . .	61
5.3. Función de Schwefel en 2D, $f(x, y) = -x \sin(\sqrt{ x }) - y \sin(\sqrt{ y })$. . .	64
5.4. Estimación de las densidades para el problema de Schewefel con $L = 100$ y $D = 50$ donde muestra las dos regiones mínimas muy bien distinguidas.	64

5.5. Comportamiento del tiempo de cómputo para el algoritmo SFPL híbrido en 50 iteraciones con respecto a la dimensión del problema de Rosenbrock.	67
5.6. Comportamiento del tiempo de cómputo para el algoritmo SFPL híbrido en 100 iteraciones con respecto a la dimensión del problema de Rosenbrock.	68
5.7. Comportamiento del tiempo de cómputo para el algoritmo SFPL híbrido en 200 iteraciones con respecto a la dimensión del problema de Rosenbrock.	68
A.1. Representación de una combinación lineal de vectores en el método del gradiente.	76

ÍNDICE DE TABLAS

4.1. Resultados de la comparación de la densidad uniforme y una densidad estimada por el SFPL para los problemas de prueba P.1 y P.2 con tolerancia de $\epsilon = 0.01$ y $\epsilon = 0.1$ respectivamente.	36
4.2. Se muestra la tasa de puntos que satisfacen a las restricciones para 10,000 soluciones del SFPL y 1,000,000 de soluciones aleatorias generadas para los problemas P.1 y P.2 con tolerancia de $\epsilon = 0.01$ en ambos problemas.	37
4.3. Lista de las 10 mejores soluciones encontradas por el algoritmo híbrido SFPL-AGII para el problema de Booth tras 100 iteraciones.	53
4.4. Comparación de los resultados de los tres algoritmo (AGS, SFPL-AGI y SFPL-AGII) para el problema de Booth. Se realizó 200 iteraciones para los dos primeros algoritmos y 100 iteraciones para el tercer algoritmo.	54
5.1. Resultados típicos en 10 corridas de los algoritmos AGS y SFPL-AGI con 50, 100 y 200 iteraciones en cada algoritmo para el problema de Booth.	60
5.2. Resultados típicos en 10 corridas de los algoritmos AGS y SFPL-AGI con 50, 100 y 200 iteraciones en cada algoritmo para el problema de Rosenbrock.	62

5.3. Resultados típicos en 10 corridas de los algoritmos AGS y SFPL-AGI con 50, 100 y 200 iteraciones en cada algoritmo para el problema de Schwefel.	65
5.4. Resultados del algoritmo genético para el problema de Rosenbrock con tamaños de dimensión de 20, 40, 60, 80 y 100.	66
5.5. Resultados del algoritmo SFPL híbrido para el problema de Rosenbrock con tamaños de dimensión de 20, 40, 60, 80 y 100.	67

AGRADECIMIENTOS

Agradezco a los profesores del PISIS por haberme orientado durante todo el tiempo que estuve como alumno y a la coordinación del programa por permitirme pertenecer a él. A la Universidad Autónoma de Nuevo León y su Facultad de Ingeniería Mecánica y Eléctrica por haberme otorgado apoyo en forma de beca de inscripción y colegiatura, respectivamente. Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por haberme otorgado beca de manutención durante la duración de la maestría, lo que hizo posible realizar mis estudios. Asimismo al proyecto CONACYT J45702-A por recursos para asistencia a congresos.

Agradecido profundamente con el Dr. Arturo Berrones por la dirección de la tesis y que ha sido para mi una persona que me ha apoyado, orientado, aconsejado, brindado el tiempo para resolver mis dudas y que además me ha comprendido, motivado a continuar con mis estudios y transmitido ese ánimo para trabajar y formarme como investigador.

Agradecido también con la Dra. Satu Elisa Schaeffer por su ayuda incondicional en todo momento que me brindo durante la maestría y que me aportó un amplio conocimiento en herramientas computacionales y lenguajes de programación.

A los miembros del comité de tesis Dra. Satu Elisa Schaeffer y el Dr. Roger Z. Ríos por sus valiosas recomendaciones para mejorar la calidad de la tesis y asimismo por su paciencia y comprensión.

A todos mis compañeros por brindarme su amistad y así como también por su apoyo a Miguel A. Urbano, Jaime Cano, Ricardo Sanchez, Dexmont Peña, Sergio

Madrigal, Carlos Alvarez, Eloir Blanco, Angelica Salazar y Anel Reyes. En especial a Miguel Mata por la aportación de la plantilla de tesis en \LaTeX usada en el presente trabajo. Así como también al Dr. Javier Almager por sus valiosas recomendaciones para la mejora de la calidad de la tesis.

A mis padres por la educación que me brindaron, por su sacrificio cotidiano, por sus sabios consejos y por sus buenos ejemplos, por lo cuales no puedo hacer menos que esforzarme cada día por mi y por ellos.

A mis hermanos y mi hija Azul Zoé que siempre han estado conmigo en todo momento, y que han sido parte de mis fuerzas para seguir adelante y luchar para ser mejor y útil a quienes me rodean. En general gracias a toda mi familia por su amor y su gran cariño que me han brindado.

A mis amigos y compadres Mario Hernández Reyes, Víctor Cervantes Velasco, Josue Alcántar Ceceña, Ramón A. López Moreno que siempre me han apoyado e impulsado a continuar con sus palabras de aliento.

Pero sobre todo, gracias a Dios, quien me ha dado la vida y me ha guiado para lograr todo lo que he deseado.

RESUMEN

Jonás Velasco Álvarez.

Candidato para el grado de Maestro en Ciencias en Ingeniería de Sistemas.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio:

UNA HEURÍSTICA BASADA EN BÚSQUEDA ESTOCÁSTICA PARA PROBLEMAS DE OPTIMIZACIÓN GLOBAL

Número de páginas: 84.

OBJETIVOS Y MÉTODO DE ESTUDIO: Se propone una heurística basada en un proceso difusivo para la estimación de densidades de probabilidad en problemas de optimización global con restricciones. El método construye expresiones analíticas para la aproximación de las densidades marginales estacionarias, en un proceso general de búsqueda estocástica. Las densidades marginales pueden ser fácilmente definidas, y se encuentran en regiones del espacio de búsqueda que con alta probabilidad contiene el óptimo global. Dado lo anterior se procede a obtener una población de puntos tomados de las densidades que, a través de un algoritmo genético, son adaptados o contraídos en las regiones donde alcanzan su mejor valor objetivo.

El procedimiento de estimación de densidad involucra un número controlado de operaciones lineales, lo que implica una gran ventaja en comparación con otros métodos. Con esto es posible especificar la complejidad de cómputo que se desea realizar.

CONTRIBUCIONES Y CONCLUSIONES: Se propone una metodología para tratar a los problemas de optimización global a gran escala con restricciones. Se contruye un algoritmo híbrido a partir del algoritmo de estimación de densidad y los algoritmos genéticos. Se propone un enfoque del algoritmo de estimación de densidad para el manejo de restricciones en algoritmos genéticos. Se presenta un primer resultado del algoritmo de estimación de densidad para problemas de optimización con incertidumbre.

Se observó que para los problemas de optimización de prueba utilizados en este trabajo, la metodología propuesta brinda soluciones de muy buena calidad en tiempo de cómputo razonable, y que dichas soluciones son consistentes con las propiedades óptimas globales conocidas.

Firma del asesor: _____

Dr. J. Arturo Berrones Santos

CAPÍTULO 1

INTRODUCCIÓN

En este primer capítulo se proporciona el objetivo, la justificación y la hipótesis del trabajo. Por último, se describe esencialmente la estructura del trabajo de tesis.

1.1 OBJETIVO

El objetivo de esta tesis es desarrollar un nuevo enfoque en el manejo de restricciones de los algoritmos de estimación de densidad para tratar a problemas de optimización global. Asimismo, se pretende tener un algoritmo híbrido a partir del algoritmo de estimación de densidad y los algoritmos genéticos con el fin de obtener soluciones adaptadas o contraídas en regiones del espacio de búsqueda donde se tiene esencialmente configuraciones que reflejen la solución óptima global, con un esfuerzo computacional razonable. Además, se pretende construir un enfoque del algoritmo de estimación de densidad como un mecanismo para el manejo de restricciones en los algoritmos de genéticos. Por último, se desea evaluar el potencial del algoritmo híbrido en un problema que crece en dimensión.

1.2 JUSTIFICACIÓN

Es importante tener un método con el cual mediante una búsqueda eficiente se estima una densidad que refleja la información aprendida en el proceso difusivo. Con esta estimación de densidad se obtiene un punto inicial o una población de puntos

que provienen de regiones cercanas al óptimo global y que pueden ser brindados a los algoritmos deterministas o estocásticos. En otras palabras, ésta es una heurística que enlazada a otras heurísticas o algoritmos exactos, serviría como un mecanismo de diversificación que les ayudaría en gran medida a encontrar soluciones óptimas de manera eficiente. Asimismo en este método se tiene la completa libertad de decidir el esfuerzo computacional mediante los parámetros libres para la estimación.

Por otro lado, el algoritmo de estimación de densidades es importante ya que sirve como un manejador de restricciones en métodos heurísticos como los algoritmos evolutivos entre otros, para ayudar a encontrar soluciones a problemas de optimización. Con esto último se tiene un importante avance en el estado del arte de los algoritmos deterministas y estocásticos dedicados a resolver un problema de optimización global.

1.3 HIPÓTESIS

Se tiene una metodología para tratar problemas de optimización global con restricciones, en donde el algoritmo de estimación de densidad brinda un punto o una población de puntos dados en las regiones cercanas al óptimo global y que esas soluciones son adaptadas mediante un algoritmo genético en la región óptima global de manera eficiente en un tiempo de cómputo razonable.

1.4 ESTRUCTURA DE LA TESIS

La estructura del presente trabajo de tesis ha sido elaborada con el objetivo de crear un cuerpo comprehensivo, incluso para los lectores menos familiarizados con los temas que se tratan. Con esta intención se presenta en los apéndices material que consideramos de utilidad para la mejor comprensión del presente trabajo.

La tesis ha sido organizada en siete capítulos. A continuación se hace una descripción de lo que se encuentra en cada uno de ellos.

En el capítulo segundo se presentan los antecedentes a la metodología propuesta y algunas comparaciones de otras metodologías con la nuestra.

En el capítulo tercero se describe la metodología, el algoritmo de estimación de densidad, los parámetros del algoritmo, la representación gráfica de las densidades, la convergencia de las densidades hacia un estado estacionario, así como la aplicación del método para estimar las densidades de un problema de optimización sin restricciones y de un problema con incertidumbre.

En el capítulo cuarto se presenta el estudio del algoritmo de estimación de densidades en problemas con restricciones, donde se discuten diferentes formas para el manejo de las restricciones. Se presenta una breve introducción a los algoritmos genéticos. Asimismo se describe el algoritmo genético modificado para nuestra metodología. Se propone un enfoque de hibridación del algoritmo propuesto con los algoritmos genéticos. Por último, se discute un enfoque del algoritmo de estimación de densidades como un mecanismo para el manejo de restricciones en algoritmos genéticos.

En el capítulo quinto se presentan los detalles de la experimentación computacional y resultados. Se presenta un análisis estadístico de los resultados típicos obtenidos por el algoritmo híbrido propuesto y el algoritmo genético. Asimismo se realiza una comparación del desempeño del algoritmo híbrido contra los algoritmos genéticos en un problema que crece en dimensión, con el fin de evaluar el comportamiento del tiempo de cómputo con respecto al tamaño del problema y la calidad de la solución. Por último se exponen comentarios de los resultados obtenidos.

En el capítulo sexto se presentan un resumen de las contribuciones realizadas, las conclusiones generales del trabajo, así como sugerencias de posibles trabajos futuros.

CAPÍTULO 2

ANTECEDENTES

Uno de los algoritmos heurísticos básicos de búsqueda estocástica en optimización y de aplicación general, es el algoritmo de búsqueda ciega [34], que escoge soluciones aleatorias de manera uniforme sobre un espacio de búsqueda dado. La búsqueda ciega suele ser eficiente en problemas de pequeñas dimensiones. En problemas que implican una mayor dimensión este método se vuelve muy ineficiente debido a que el espacio de búsqueda en cualquier intervalo real, contiene un número infinito de soluciones y la probabilidad de escoger cualquier valor entre ese intervalo tiende a cero [34], por lo que encontrar una solución óptima en un tiempo de cómputo razonable no sería posible. Dado que este algoritmo trabaja mediante una búsqueda exhaustiva, una de las formas de ayudar a realizar la búsqueda eficientemente es tomar un subintervalo del intervalo dado, con lo cual la probabilidad de escoger un valor en ese subintervalo es mayor a cero. De acuerdo con lo anterior, la probabilidad de encontrar una solución crece o disminuye dependiendo del tamaño del subintervalo. Para el caso de realizar una búsqueda ciega en N variables, esto implicaría una geometría y un espacio de búsqueda más complicado.

Un algoritmo más eficiente que la búsqueda ciega y de implementación muy sencilla, se basa en considerar la información del gradiente en la función objetivo. El método del gradiente converge a encontrar soluciones óptimas locales [34]. Algo muy útil para obtener diferentes soluciones óptimas locales en el método del gradiente, es probar con varios puntos iniciales, es decir diferentes condiciones de inicio para poder diversificar en el proceso de búsqueda. Este enfoque se le conoce como un método de

multi-inicio o multi-arranque. Otra forma para realizar una búsqueda eficiente con el método del gradiente, es que se generalice mediante la inclusión de un término estocástico de diversificación. Esto se conoce como búsqueda difusiva o proceso difusivo [33, 37]. Estos procesos difusivos tienen una densidad de probabilidad asociada la cual refleja la información obtenida en el proceso de búsqueda. Dicha información nos muestra las regiones donde se localizan los puntos óptimos locales de todo el espacio de interés.

2.1 ALGORITMOS DE BÚSQUEDA ESTOCÁSTICA EN LA OPTIMIZACIÓN GLOBAL

Las estrategias estocásticas para la optimización, son esencialmente usadas en la mayoría de las técnicas heurísticas, para enfrentar a los sistemas complejos, sean estos los problemas de optimización global no estructurados [29]. La raíz de tales métodos se remonta al algoritmo de metrópolis [23], que se introdujo en épocas tempranas de la computación científica. Dicho algoritmo simula la evolución de un sistema físico hacia un equilibrio termal. El método está basado en la técnica del recocido simulado [19]. El recocido simulado es una analogía de un sistema físico, que se basa en el proceso natural de enfriamiento lento de un sólido, donde la temperatura tiende hacia cero hasta que alcanza su punto de equilibrio que lo lleva a la energía mínima global.

El método presentado en este trabajo está basado en principios físicos similares, como el algoritmo de recocido simulado [19, 30, 35]. Sin embargo, a diferencia con este enfoque, el método propuesto considera una densidad de puntos en vez de una transición de Markov de puntos individuales. La técnica propuesta está basada en la interacción entre las ecuaciones de Langevin y Fokker–Planck para procesos estocásticos, las cuales son muy conocidas para el estudio de sistemas físicos que se encuentran fuera de equilibrio [33, 37]. La ecuación de Fokker–Planck ya ha sido utilizada en diversas aplicaciones de algoritmos de búsqueda, por ejemplo, en el estudio

de la convergencia de poblaciones de puntos a un mínimo global [35]. Asimismo ha sido utilizada como herramienta para demostrar la convergencia de los algoritmos tipo recocido simulado [11, 30], o como una estructura teórica para maquinas de aprendizaje tipo Boltzmann [20, 26]. En optimización global, la ecuación de Fokker–Planck también ha sido utilizada en el contexto de poblaciones de puntos como un esquema en donde las poblaciones evolucionan sobre versiones de la ecuación que son dependientes del tiempo y que sigue un programa para la reducción de la constante de difusión D [35].

En el presente trabajo la versión estacionaria de la ecuación de Fokker–Planck se usa para el aprendizaje de la densidades de probabilidad estacionarias en un proceso de búsqueda estocástica general, las cuales se obtienen con un número relativamente pequeño de evaluaciones de una función de costos dada en un tiempo de cómputo razonable.

CAPÍTULO 3

ESTIMACIÓN DE DENSIDADES DE PROBABILIDAD

3.1 DENSIDADES DE PROBABILIDAD EN UN PROCESO DE BÚSQUEDA ESTOCÁSTICA

La optimización de una función de costos la cual presenta un gran número de mínimos locales es un tema relevante en los campos de la ciencia y la ingeniería. Para encontrar la solución mínima global eficientemente, se requiere seleccionar una solución inicial muy buena o un conjunto de soluciones iniciales. En los algoritmos deterministas como en los estocásticos para la optimización continua no lineal, tener ese conocimiento es de gran importancia. En este trabajo se propone una técnica para determinar puntos iniciales que estén en regiones cercanas al óptimo global o, dicho de otra forma, puntos que se encuentren en las regiones que con alta probabilidad contienen al punto óptimo global.

Esta técnica estocástica ha sido introducida recientemente por Bagnoli et al. [1], la cual está relacionada con el área de procesos estocásticos en donde se utilizan las ecuaciones de Langevin y la de Fokker-Planck, y que son conocidos en los sistemas de estudio de sistemas físicos fuera de equilibrio [33, 37]. Dada una función de costo $V(x_1, x_2, \dots, x_n, \dots, x_N)$, una búsqueda estocástica puede ser modelada por la ecuación de Langevin,

$$\dot{x}_n = -\frac{\partial V}{\partial x_n} + \varepsilon(t), \quad (3.1)$$

donde \dot{x}_n representa al sistema dinámico de la partícula n en interacción con las demás partículas, el primer término del lado derecho de la igualdad representa al gradiente negativo como mecanismo de adaptación del proceso y $\varepsilon(t)$ es un ruido aditivo con media cero. Este es un ruido gaussiano no correlacionado con amplitud constante. Le llamaremos más adelante constante de difusión D . La constante de difusión D nos permitirá dar saltos en el proceso de búsqueda como mecanismo de diversificación. Por otro lado, el recocido simulado utiliza una difusión llamada temperatura que varía en el tiempo lentamente para llegar al equilibrio térmico [19]. La ecuación (3.1) se puede interpretar como un sistema dinámico compuesto de N partículas que interactúan entre ellas y nos describe la evolución no lineal para el estado \vec{x} y se representa gráficamente en la Figura 3.1.

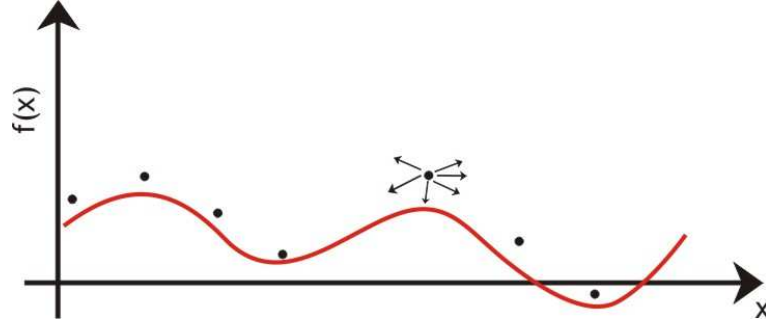


Figura 3.1: N partículas que interactúan en un potencial

Con esto la densidad de probabilidad asociada al proceso de búsqueda estocástica de la ecuación (3.1) de Fokker–Planck [33, 37] está dada por la siguiente expresión en una ecuación lineal diferencial,

$$\dot{p}(x) = \frac{\partial}{\partial x_n} \left[\frac{\partial V}{\partial x_n} p(x) \right] + D \frac{\partial^2 p(x)}{\partial x_n^2} \quad (3.2)$$

La forma de esta solución es un tipo de la bien conocida ecuación de Boltzmann [33, 37]. La densidad de probabilidad estacionaria está dada por el tiempo asintótico

de residencia de las N partículas sobre distintas regiones en el espacio después de un tiempo infinito de búsqueda. Las regiones preferidas en esta búsqueda son aquellas que minimizan el costo globalmente y se representa gráficamente en la Figura 3.2.

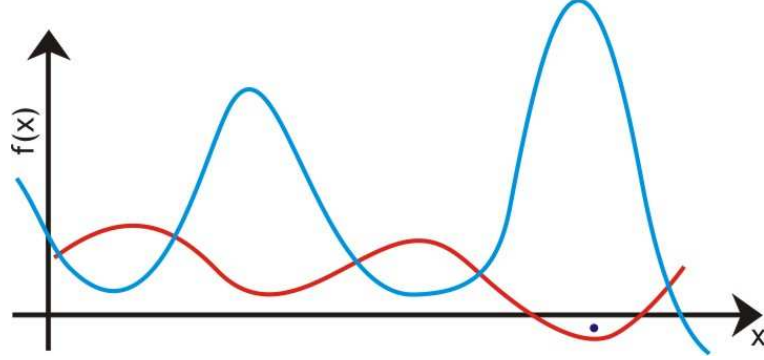


Figura 3.2: Densidad de una partícula en un potencial.

El principio fundamental de este enfoque es que para espacios de búsqueda acotados con condiciones reflejantes en los extremos, la densidad $p(x)$ converge a un estado estacionario [16]. De esta manera se puede estudiar el límite $t \rightarrow \infty$ del proceso de búsqueda (3.1) observando las soluciones estacionarias de la ecuación (3.2). Dado esto, el problema de optimización asociado se define en la región de búsqueda,

$$L_{1,n} \leq x_n \leq L_{2,n} \quad n = 1, \dots, N \quad (3.3)$$

donde $L_{1,n}$ representa la cota inferior y $L_{2,n}$ representa la cota superior o bien, el intervalo de búsqueda para x_n .

3.1.1 EL SFPL

Se llama Aprendizaje Estacionario de Fokker–Planck (SFPL por sus siglas en inglés, *Stationary Fokker–Planck Learning*) a la estimación de la densidad estacionaria de un proceso de búsqueda estocástica mediante la ecuación de Fokker–Planck.

Dado que el uso directo de la ecuación (3.2) para la optimización implica

calcular un alto número de integrales para el sistema de N dimensiones, se procede a realizar la proyección unidimensional de la ecuación (3.2) de Fokker–Planck. Esta proyección unidimensional resulta ser numéricamente mucho más fácil ya que se tiene que resolver un sistema de N ecuaciones diferenciales lineales de primer orden. Considerando la siguiente densidad condicional,

$$p(x_n|\{x_{j \neq n}\}) = \frac{p(x_1, \dots, x_N)}{\int p(\{x_{j \neq n}\}|x_n)p(x_n)dx_n}, \quad (3.4)$$

es decir, la evolución de la variable x_n dado la posición de todas las demás variables donde $p(x_n) = \int p(x_n|\{x_{j \neq n}\})p(\{x_{j \neq n}\})d\{x_{j \neq n}\}$ indica la respectiva densidad marginal.

Introduciendo la ecuación (3.4) en la ecuación (3.2), se tiene la proyección unidimensional de la ecuación (3.2) de Fokker–Planck en $t \rightarrow \infty$, $\dot{p}(x) = 0$ y está dada por,

$$D \frac{\partial p(x_n|\{x_{j \neq n} = x_j^*\})}{\partial x_n} + p(x_n|\{x_{j \neq n} = x_j^*\}) \frac{\partial V}{\partial x_n} = 0. \quad (3.5)$$

que describe la evolución de la variable x_n mientras se mantiene el resto de las variables fijas $\{x_{j \neq n} = x_j^*\}$, que se puede utilizar para sacar los puntos de la densidad condicional y con ello se tiene la muestra de la correspondiente densidad marginal $p(x_n)$. Esto es posible por la ecuación (3.5) en una ecuación diferencial de segundo orden para la distribución acumulada asociada a la densidad $p(x)$ dada por $y(x_n|\{x_{j \neq n} = x_j^*\}) = \int_{-\infty}^{x_n} p(x'_n|\{x_{j \neq n} = x_j^*\})dx'_n$ que se denota por y . Ésta se conecta a las densidades marginales por medio del muestreo de Gibbs.

En el estado estacionario la proyección unidimensional proporciona la densidad de probabilidad condicionada que corresponde al proceso N dimensional original [1]. La ecuación diferencial de segundo orden para la distribución acumulada es la siguiente,

$$\frac{d^2 y}{dx_n^2} + \frac{1}{D} \frac{\partial V}{\partial x_n} \frac{dy}{dx_n} = 0, \quad (3.6)$$

$$y(L_{1,n}) = 0, \quad y(L_{2,n}) = 1.$$

donde $y(L_{1,n})$ y $y(L_{2,n})$ son las condiciones a la frontera.

Es posible generar puntos provenientes de la densidad $p(x_n | \{x_{j \neq n} = x_j^*\})$ por el hecho de que y es una variable aleatoria distribuida uniformemente en el intervalo $y \in [0, 1]$, lo cual se sigue de las leyes de transformación de funciones de probabilidad [1]. De acuerdo con los principios de muestreo de Gibbs la iteración de los pasos previos sobre las N variables producirá una población que es muestreada de las correspondientes densidades marginales $p(x_n)$. Al ver a x_n como una variable aleatoria, $y(x_n | \{x_{j \neq n}\})$ puede ser aproximada mediante una combinación lineal de funciones de un conjunto completo que satisfaga las condiciones de los extremos en el intervalo de interés. Por consiguiente, la estimación de la forma analítica de $y(x_n)$ puede conseguirse por la sustitución de la expansión,

$$\hat{y}(x_n | \{x_{j \neq n}\}) = \sum_{l=1}^L a_{n,l} \varphi_l(x_n), \quad (3.7)$$

donde $a_{n,l}$ son los coeficientes de la expansión y $\varphi_l(x_n)$ es la función base usada para la aproximación.

El método SFPL está basado en la representación de condicionales acumuladas $\hat{y}(x_n | x_{j \neq n})$ de una solución estacionaria.

Eligiendo, por ejemplo, una base en la cual $\varphi_l(0) = 0$, uno de los coeficientes queda determinado. Los $L - 1$ coeficientes restantes se obtienen resolviendo la ecuación (3.6) para $L - 1$ valores de x_n en sus intervalos de interés. De esta manera la aproximación de $y(x_n)$ se realiza resolviendo un conjunto de L ecuaciones algebraicas para los coeficientes a 's, iniciando en un punto arbitrario para x_n en el

dominio y se procede a desarrollar la expansión (3.7), teniendo con esto la siguiente ecuación. Del resultado de la marginal de x_n , un nuevo punto es muestreado para obtener la siguiente ecuación. Este muestreo representa una versión particular de un muestreo de Gibbs [10]. De esta forma se procede a obtener los coeficientes a 's, los cuales ahora tienen la información sobre la función objetivo V . Esto implica $L - 1$ evaluaciones de la derivada V .

Bajo condiciones generales [10], los puntos generados mediante el muestreo de Gibbs a partir de las densidades condicionales de una variable, están asintóticamente distribuidos de acuerdo con las marginales de la distribución conjunta de N variables. La convergencia del SFPL a la distribución en equilibrio proviene del hecho que el procedimiento representa un muestreo de Gibbs [10].

En nuestro formalismo, es posible no sólo muestrear, sino ofrecer una expresión analítica aproximada de la densidad marginal:

$$\langle \hat{y} \rangle = \sum_{l=1}^L \langle a_{n,l} \rangle \varphi_l(x_n) \rightarrow y(x_n), \quad (3.8)$$

donde los corchetes representan el promedio sobre el muestreo. Por lo tanto con $\langle \hat{y} \rangle$ se obtiene una aproximación de la densidad $p(x_n)$, la cual representa las densidades en equilibrio para la variable x_n después de un tiempo infinito de búsqueda de un proceso de búsqueda estocástica. La densidad obtenida puede ser usada para graficar una población de puntos en las regiones óptimas globales.

Para los experimentos, la siguiente forma específica de la expansión (3.7) se ha utilizado,

$$\hat{y} = \sum_{l=1}^L a_l \sin \left((2l-1) \frac{\pi(x_n - L_{1,n})}{2(L_{2,n} - L_{1,n})} \right). \quad (3.9)$$

Es importante mencionar que la elección de las funciones base es bastante arbitraria, por lo que otro conjunto completo de funciones base podría ser considerado

(e.g. polinomios de Hermite). Se usa la base de Fourier (base senoidal) solo porque tiene relativa simplicidad [21].

3.1.2 EL ALGORITMO SFPL

Se muestra a continuación el algoritmo de Aprendizaje Estacionario de Fokker–Planck con mayor precisión.

Algoritmo 3.1 $SFPL(M, Nvar, L_1, L_2)$

Input: M : número de iteraciones del algoritmo; $Nvar$: número de variables del problema; $[L_1, L_2]$ intervalo del dominio de las variables.

Output: $\bar{a}_{n,l}$: promedio del coeficiente $a_{n,l}$ sobre las iteraciones, $\forall n = 1, \dots, Nvar$, $l = 1, \dots, L$.

$x_j \leftarrow \text{Dist_unif}(L_1, L_2) \forall j \in 1, \dots, Nvar$;

$\bar{a}_{n,l} \leftarrow 0 \forall n = 1, \dots, Nvar, l = 1, \dots, L$;

for $i = 1, \dots, M$

for $n = 1, \dots, Nvar$

$a_{n,l} \leftarrow \text{Estimar_coeficientes}(x, n)$;

$x_n \leftarrow \text{Obtener } \hat{y}(x_n)$;

end for

$\bar{a}_{n,l} \leftarrow \bar{a}_{n,l} + a_{n,l}, n = 1, \dots, Nvar, l = 1, \dots, L$;

end for

$\bar{a}_{n,l} \leftarrow \frac{\bar{a}_{n,l}}{M}, n = 1, \dots, Nvar, l = 1, \dots, L$;

return $\bar{a}_{n,l}$;

3.1.3 PARÁMETROS DEL ALGORITMO SFPL

El algoritmo de estimación de densidad consta de dos parámetros muy importantes con los cuales se puede estimar la densidad con gran precisión y controlar el

costo computacional requerido en la estimación.

El primer parámetro es D , que representa la difusión que es proporcional a la intensidad o la variabilidad de ruido aditivo en el proceso de búsqueda. Se puede notar que si en la ecuación (3.1) el parámetro D tiene una amplitud infinita, la influencia dinámica de la función de costos sobre el proceso de búsqueda se perdería, por lo tanto el proceso se convierte en una búsqueda ciega. Por otro lado si D tiene una amplitud de ruido que esté variando lentamente en el tiempo, tendremos un proceso muy similar al del recocido simulado. De otra forma, si el parámetro D es constante en el proceso de búsqueda, este sería un proceso por pura difusión. Si D es constante y de tamaño pequeño, el proceso de búsqueda sería muy preciso para encontrar regiones que con alta probabilidad se localiza el óptimo global, por lo que esto también implica un mayor costo computacional del algoritmo.

El segundo parámetro, L , está asociado al número de evaluaciones del gradiente en la función objetivo como se puede notar en la ecuación (3.7). Esto representa la precisión con la cual se aproxima la densidad a estimar. Por lo tanto si L es grande se tiene una densidad muy aproximada aunque genera un mayor costo computacional en la estimación.

Como se puede notar, el tener la libertad de manejar los parámetros para estimar las densidades nos ayuda a controlar el costo computacional con sólo el ajuste apropiado de los mismos. El tamaño de los parámetros varía con cada problema y éstos dependen de las características y estructuras en los espacios de búsqueda de los mismos. Los parámetros apropiados deben describir distribuciones correctas para su validez. Una distribución correcta está dada cuando las densidades estimadas no contienen valores negativos al ser graficadas. Por lo tanto, en cualquier densidad la distribución asociada siempre tiene características de ser creciente, biunívoca y acotada entre cero y uno. A través de la estimación de la densidad se pueden conocer también las distribuciones de cada variable por lo que se puede verificar que éstas sean correctas.

3.2 EL SFPL PARA PROBLEMAS SIN RESTRICCIONES

El algoritmo SFPL anteriormente ha sido probado en problemas sin restricciones en [2, 3, 31], en problemas donde se tiene una sola función de términos no lineales y un espacio de búsqueda definido o acotado para las variables x_n . Para tales problemas se han estimado satisfactoriamente las densidades asociadas al proceso de búsqueda global. Para fin de mostrar el trabajo de SFPL, en esta sección se muestra cómo el SFPL nos estima las densidades para cada variable. Asimismo se muestra la convergencia de las densidades hacia el estado estacionario a través de un número determinado de iteraciones para el SFPL.

El problema sin restricciones es modelado de la siguiente forma,

$$\begin{aligned} & \text{mín} \quad f(x) \\ & \text{sujeto a} \quad L_{1,n} \leq x_n \leq L_{2,n} \quad n = 1, \dots, N \end{aligned} \tag{3.10}$$

donde $f(x)$ es una función objetivo arbitraria con términos no lineales. Se puede ver que el SFPL se plantea como un problema de minimización. En caso de un problema de maximización, se puede transformar a uno de minimización mediante la relación $\text{máx} \quad f(x) = - \text{mín} \quad [-f(x)]$.

3.2.1 DENSIDADES DE LAS FUNCIONES

En esta sección se ilustran las densidades estimadas para cada variable que se obtienen mediante el SFPL. Para ilustrar estas densidades se utiliza el software Grace en Ubuntu 7.10 [12] con las cuales son graficadas.

A continuación se presenta la función de Booth [22] que es muy usada como función de prueba para evaluar el desempeño de algoritmos heurísticos. Por ende se utiliza ésta para evaluar el desempeño del SFPL, y para ilustrar las densidades

estimadas que describen la región donde se localiza la solución óptima global con alta probabilidad. La función se describe de la siguiente forma,

Función de Booth:

$$\begin{aligned} \text{mín} \quad & f(x_1, x_2) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2, \\ \text{sujeta a} \quad & -10 \leq x_1 \leq 10, \\ & -10 \leq x_2 \leq 10, \end{aligned} \quad (3.11)$$

Solución: $x^* = (1, 3)$, $f(x^*) = 0$.

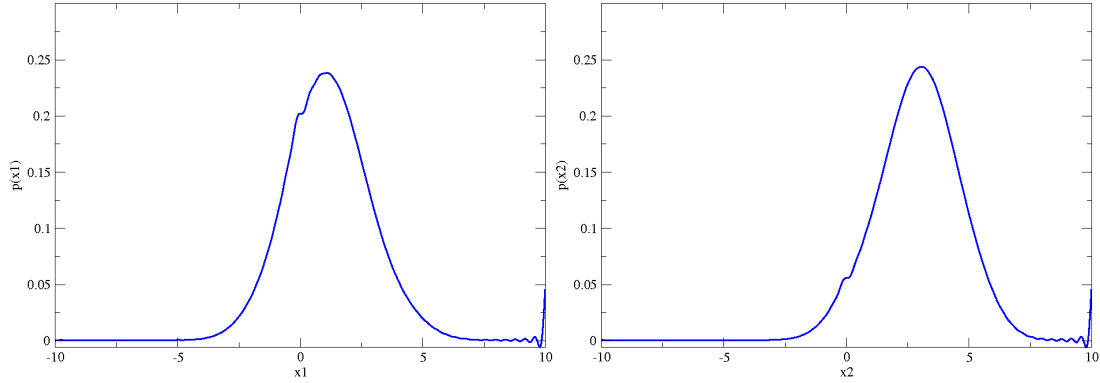


Figura 3.3: Estimación de las densidades de x_1 y x_2 para la función de Booth, usando parámetros de $L = 100$ y $D = 10$.

En la Figura 3.3 se ilustran las densidades estimadas para las variables x_1 y x_2 de la función de Booth. Las densidades estimadas se localizan en la región donde se encuentra la solución óptima conocida. Para la función de Booth el algoritmo SFPL estima la densidad en la región donde esta contenida la solución óptima global.

3.2.2 CONVERGENCIA DE LAS DENSIDADES DEL SFPL

Esta sección se divide en dos partes. En la primera se ilustra la convergencia de las densidades hacia una correcta distribución para el problema de Booth y en la segunda parte se ilustra la convergencia de las densidades en un problema con incertidumbre. Se utilizan diferentes iteraciones del SFPL con el fin de mostrar las

densidades en cada una de ellas, y con esto ilustrar densidades hacia un estado estacionario en el proceso de búsqueda.

CONVERGENCIA DE LAS DENSIDADES DE LA FUNCIÓN DE BOOTH

Con el fin ilustrar la convergencia de las densidades en un problema sin restricciones se hace uso nuevamente del la función de Booth.

Se puede ver en la Figura 3.4 como las densidades iteración tras iteración convergen a un estado estacionario. En el experimento realizado, la iteración 5 se tiene que el punto máximo de la densidad es de $x = (2.17, 1.95)$, para la iteración 50 el punto máximo es de $x = (0.769, 3.22)$ y en la iteración 100 el punto máximo de la densidad es de $x = (1.09, 3.08)$. La solución conocida del problema es $x^* = (1, 3)$ en sus respectivas variables. Con esto se determina que la región de alta probabilidad es totalmente consistente con la solución conocida y la densidad iteración tras iteración converge efectivamente a un estado estacionario definiendo mejor la región donde se encuentra el óptimo global.

CONVERGENCIA DE LAS DENSIDADES PARA UN PROBLEMA CON INCERTIDUMBRE

Debido a su formulación estocástica, el algoritmo SFPL aparece naturalmente para ser adaptado en problemas que envuelven parámetros con incertidumbre en la función de costos. En primera instancia, se prueba el desempeño del SFPL en el conocido problema del vendedor de periódico para la cual se conoce una expresión general de la solución. Esta solución se utiliza para comparar la calidad de la estimación dada por el SFPL.

El modelo se usa para determinar los niveles de inventario óptimo. Suponemos que x es la cantidad de materia prima en el almacén de un centro de producción. El parámetro q dado es el precio de venta de cada materia prima transformada en un producto dado. El parámetro c representa el costo de producción por unidad de materia prima. El parámetro d es una demanda aleatoria, distribuida de acuerdo a

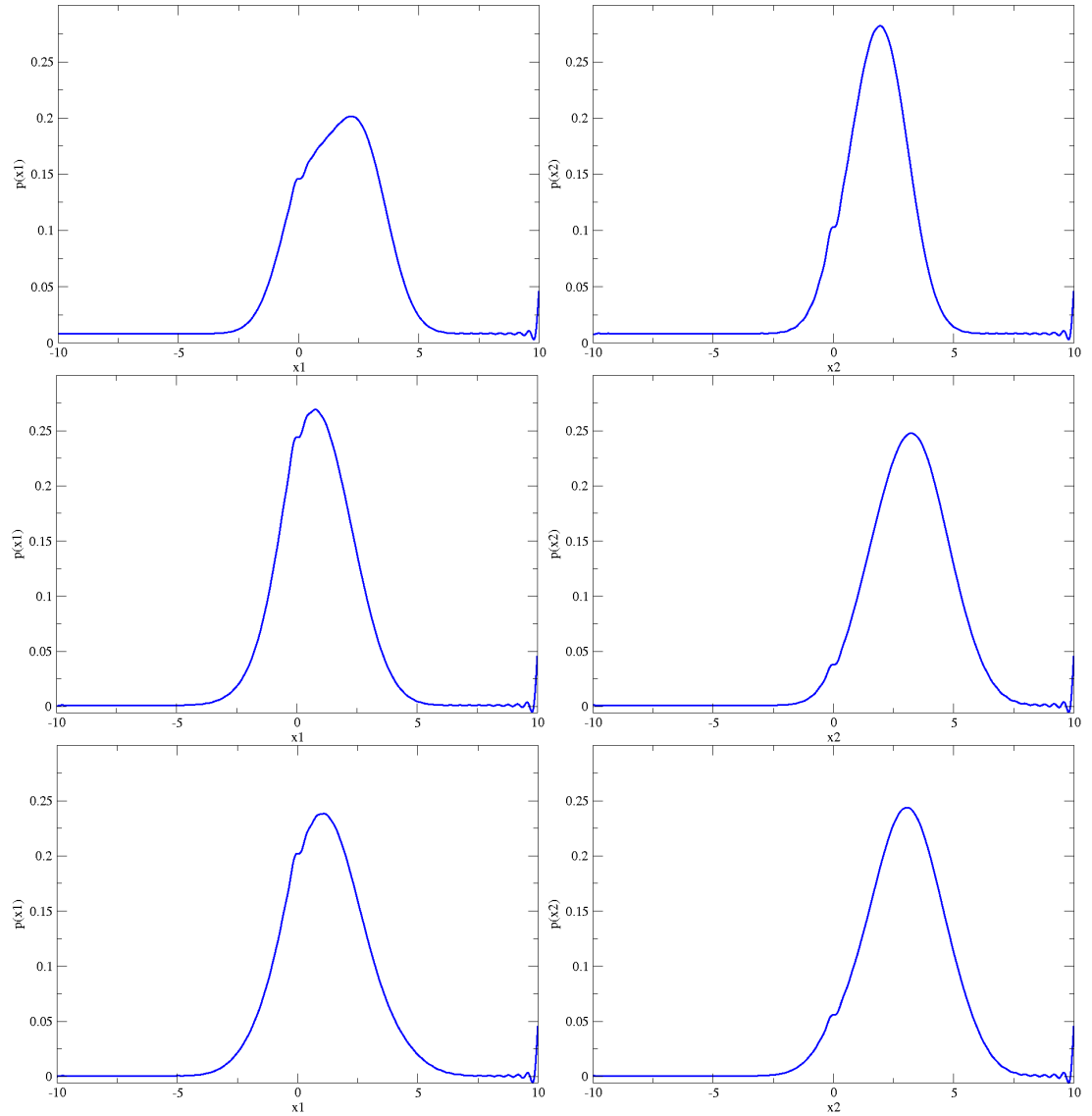


Figura 3.4: Convergencia de las densidades a una correcta distribución al estado estacionario en la función de Booth para x_1 y x_2 con 5, 50 y 100 iteraciones del SFPL. Se utilizó los parámetros de $L = 100$ y $D = 10$.

alguna distribución acumulada $F(d)$. El problema es encontrar una x , en el sentido de maximizar la ganancia esperada.

El problema del vendedor de periódico es formulado de la siguiente forma [4],

$$\max_x \{E[q - \min(x, d)] - cx\}. \quad (3.12)$$

La solución analítica del problema es dada por [4],

$$x = F^{-1} \left(\frac{q - c}{q} \right), \quad (3.13)$$

donde F^{-1} denota la inversa de la distribución acumulada de d . Para aplicar el método de SFPL al problema, se considera la formulación equivalente siguiente,

$$\min_x \{cx - E[q - \min(x, d)]\}. \quad (3.14)$$

Un experimento con una demanda con valores dados de una distribución uniforme se ilustra en la Figura 3.5. Para cada evaluación del gradiente se pasa una demanda d fija a la función de costo, la cual es evaluada por el método propuesto a continuación.

Algoritmo 3.2 *Eval_func_costos*(x , d)

Input: x : un punto del dominio, d : una demanda fija.

Output: Un valor de la evaluación en la función de costos.

if ($x < d$) $valor \leftarrow cx - qx$;

else $valor \leftarrow cx - qd$;

return $valor$;

La Figura 3.5 indica la convergencia a una correcta distribución en el problema con $p = 7$, $c = 5$ y una demanda distribuida uniformemente en el intervalo $[50, 80]$. La región de alta probabilidad es muy consistente con la solución conocida del problema, dada por $x^* = 58.55$.

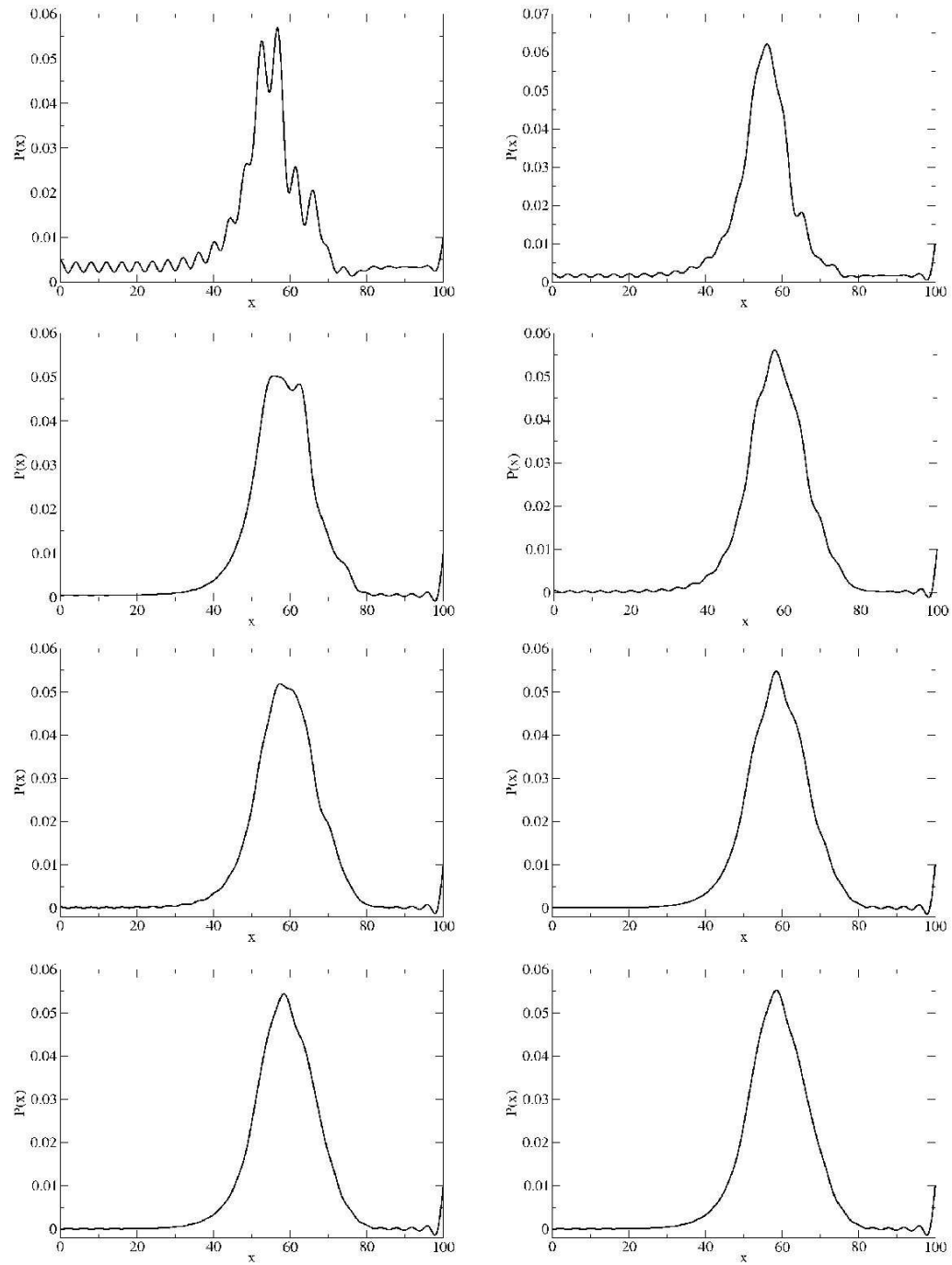


Figura 3.5: Se ilustra la convergencia de las densidades estimadas por el SFPL para el problema del vendedor de periódico. Las iteraciones del SFPL para x son de 2, 5, 20, 50, 100, 200, 500, 800 de izquierda a derecha. Se utilizaron los parámetros $L = 50$ y $D = 10$.

3.3 CONCLUSIONES DEL CAPÍTULO

Se puede notar que para el problema sin restricciones y el problema con incertidumbre el SFPL estima las densidades que describen las regiones que con alta probabilidad contiene el óptimo global. Como se puede notar en ambos problemas estas regiones son consistentes con la solución óptima del problema. Por otro lado, también se ilustró la convergencia de las densidades hacia el estado estacionario en diferentes iteraciones del algoritmo SFPL en los dos problemas que se presentaron. Se observó que el algoritmo SFPL converge a la densidad estacionaria del proceso de búsqueda estocástica, que refleja la región donde se encuentra la solución óptima global.

El potencial del SFPL nos sirve como herramienta para describir regiones que son muy consistentes con la solución óptima conocida a través de estimar las densidades. Esto nos da un indicio de que el SFPL puede ser una herramienta muy prometedora para tratar a problemas con incertidumbre así como también a problemas sin restricciones.

CAPÍTULO 4

ESTUDIO DEL SFPL EN PROBLEMAS CON RESTRICCIONES

4.1 ANTECEDENTES

Para la introducción de las restricciones en el SFPL se usa el término de «regiones de baja probabilidad». Esto significa que, durante las iteraciones del algoritmo SFPL las densidades marginales deben aprender un valor asociado con baja probabilidad para las regiones que se encuentran fuera del dominio válido. Esto se implementa de varias formas. Por ejemplo, un término no lineal apropiado puede agregarse a la función de costos. La utilidad de este enfoque, ya ha sido discutido por Berrones [3]. Por otro lado, se maneja otro enfoque en el cual se introducen restricciones a través de la construcción de barreras en términos de funciones no lineales suaves. Esto conduce a la introducción de un número de parámetros libres adicionales que controlan la forma de la no linealidad.

4.2 MÉTODO DE PENALIZACIÓN

Para el uso del SFPL en problemas con restricciones se propuso considerar funciones con un término de penalización suave y continua en la frontera. En el enfoque del SFPL para problemas restringidos, las restricciones entran a la función de costos en forma muy directa. Dicha función nos da una baja probabilidad necesaria en

las regiones donde no se satisfacen tales restricciones con la ayuda de los parámetros más o menos libres. Este enfoque en primera instancia se prueba en el clásico problema de la mochila, clasificado en la literatura como un problema NP-duro [28, 18], que se describe a continuación.

4.2.1 EL PROBLEMA DE LA MOCHILA

El problema de la mochila (KP por sus siglas en inglés, *Knapsack Problem*) que de forma general se refiere a el empaqueo de objetos dentro de una mochila, es un problema de optimización clásico que se ha abordado con varias técnicas heurísticas y exactas para resolverlo. En esta sección se propone el SFPL para estimar densidades que reflejen las configuraciones de solución óptima. El problema de la mochila se define formalmente como sigue. Sea $J = \{1, \dots, N\}$ un conjunto de objetos donde el objeto n tiene un valor q_n y un peso w_n , $n \in J$. La capacidad de la mochila está dada por C . El objetivo es seleccionar un subconjunto de J tal que la ganancia total de esos objetos seleccionados es maximizado y el total de los pesos no excede a C [18, 32].

El problema de la mochila original se formula matemáticamente de la siguiente forma,

$$\begin{aligned} & \text{máx} \quad \sum_{n=1}^N q_n x_n \\ & \text{sujeta a} \quad \sum_{n=1}^N w_n x_n \leq C, \\ & \quad x_n \in \{0, 1\} \quad n \in J \end{aligned} \tag{4.1}$$

donde x_n es la variable de decisión, que toma valor de 1 en el caso de incluir el objeto en la mochila y 0 en el caso de no incluirlo.

Con el fin de utilizar el SFPL, una transformación equivalente del problema es dada de la siguiente forma,

$$\begin{aligned}
& \min \quad - \sum_{n=1}^N q_n x_n \\
& \text{sujeta a} \quad \sum_{n=1}^N w_n x_n \leq c, \\
& \quad \quad \quad -x_n^2 + x_n \leq 0, \\
& \quad \quad \quad 0 \leq x_n \leq 1, \quad n = 1, \dots, N
\end{aligned} \tag{4.2}$$

La restricción cuadrática es equivalente a la usual restricción para variables binarias. La restricción no lineal se modela por una barrera en términos de una función sigmoïdal,

$$\begin{aligned}
& \min \quad - \sum_{n=1}^N q_n x_n + k_0 \sum_{n=1}^N \frac{1}{1 + \exp(-b_0[-x_n^2 + x_n])} \\
& \text{sujeta a} \quad \sum_{n=1}^N w_n x_n \leq c, \\
& \quad \quad \quad 0 \leq x_n \leq 1, \quad n = 1, \dots, N
\end{aligned} \tag{4.3}$$

donde k_0 y b_0 son los parámetros libres que controlan la forma de la no linealidad, para la construcción de la barrera en términos de funciones no lineales suaves.

La formulación (4.3) aproxima al modelo de la mochila entero por una función de costo no lineal suave con una restricción lineal. La restricción lineal es manejada por un término suave de la siguiente forma,

$$\min \quad Q \quad \text{sujeta a} \quad 0 \leq x_n \leq 1, \tag{4.4}$$

donde

$$Q = \begin{cases} V(x_1, x_2, \dots, x_n, \dots, x_N) & \text{si } \sum_{n=1}^N w_n x_n > c, \\ -\sum_{n=1}^N q_n x_n + k_0 \sum_{n=1}^N \frac{1}{1 + \exp(-b_0[-x_n^2 + x_n])} & \text{de otro modo.} \end{cases}$$

El aprendizaje de las distribuciones marginales estacionarias solo involucran gradientes, los cuales son necesarios para evaluar las restricciones en cada punto interior en donde la ecuación (3.6) es evaluada. Si para un punto interior se viola la restricción, el gradiente de Q es dado por el gradiente de V , que se puede construir en lugar de calcularse a partir de una definición específica de R . De esta manera se utiliza la siguiente forma,

$$\frac{\partial V}{\partial x_n} = \exp(w_n x_n - c/N) - 1.0, \quad (4.5)$$

la que da una fuerza con crecimiento exponencial que es atractiva a la región en la cual la restricción se satisface.

Para propósito ilustrativo, considerar $q = (2, 3, 5)$, $w = (3, 5, 7)$, $c = 10$ de un problema de la mochila. Por inspección, la solución es dada por $x^* = (1, 0, 1)$. En la Figura 4.1 se muestra las densidades típicas producidas por el algoritmo de estimación para esa instancia.

En la Figura 4.1 se puede notar que las densidades estimadas para cada variable describen la configuración que corresponde a la solución óptima global en la que se tiene la máxima probabilidad. La solución exacta se calcula con el algoritmo de branch and bound en el GLPK (por sus siglas en inglés, *GNU Linear Programming Kit*) [13] para problemas de programación lineal.

Con el fin de ilustrar las densidades estimadas para el problema de la mochila con 10 variables, ahora se va a considerar $q = (2.19722, 7.83202, 4.62896, 8.77338, 2.6164, 3.1601, 2.88804, 4.76096, 2.57599, 5.03961)$, $w = (1.28726, 6.92915, 3.73238,$

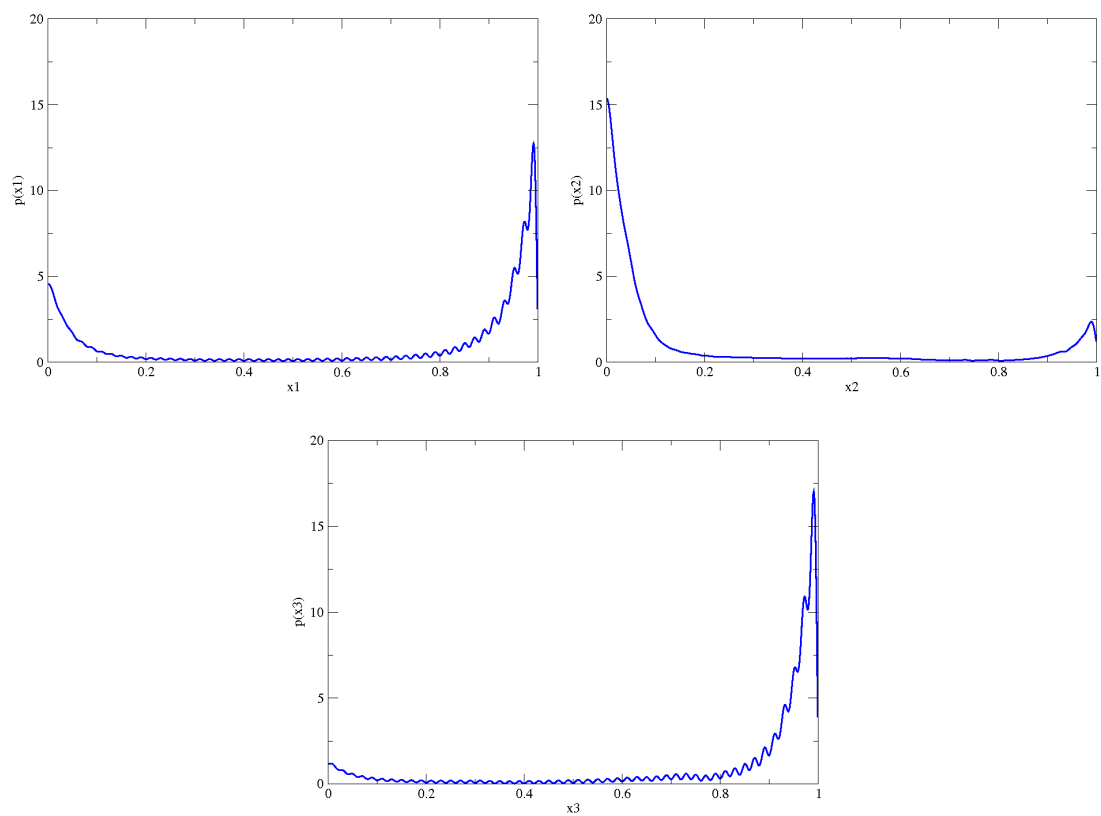


Figura 4.1: Estimación de las densidades de probabilidad para el problema de la mochila de tres variables con parámetros para el SFPL de $L = 100$ y $D = 1$.

7.89479, 1.7776, 2.19741, 2.05433, 3.88989, 1.68603, 4.10843), $c = 10$ y la solución dada por el procedimiento exacto de GLPK es $x^* = (1, 0, 0, 0, 1, 1, 1, 0, 1, 0)$.

Como se puede apreciar en la Figura 4.2, para este problema de diez variables las densidades que se estimaron corresponde a la solución óptima global conocida. Esto se puede observar comparando la configuración de la solución exacta con la configuración de la solución reflejada en las densidades para cada variable. La región de máxima probabilidad en las densidades estimadas indican la solución al problema.

Desafortunadamente, el método de penalización dado no fue suficientemente general, ya que si bien para los problemas de 3 y 10 variables el método de penalización pudo estimar las densidades muy bien, cuando las instancias fueron más grandes las densidades no correspondieron correctamente a la configuración exacta del problema.

Por esta razón, se procede a la búsqueda de otros factores de penalización. En la literatura se proponen diversos factores de penalización y funciones de reparo, las cuales no son formas generales para tratar a los problemas ya que dependen de las características del problema para una solución muy aproximada [15, 25, 24]. Otros factores de penalización requieren de muchos parámetros de tal suerte que para obtener buenos resultados se debe realizar mucha experimentación para tener los adecuados al problema específico. Asimismo se requiere de funciones que tengan propiedades de convexidad.

Por lo mencionado anteriormente, se propone entonces un nuevo enfoque para el manejo de las restricciones en algoritmos heurísticos a través del SFPL. Aquí se proponen diferentes formas de para trabajar las restricciones en problemas de optimización, que brindan soluciones de calidad eficientemente.

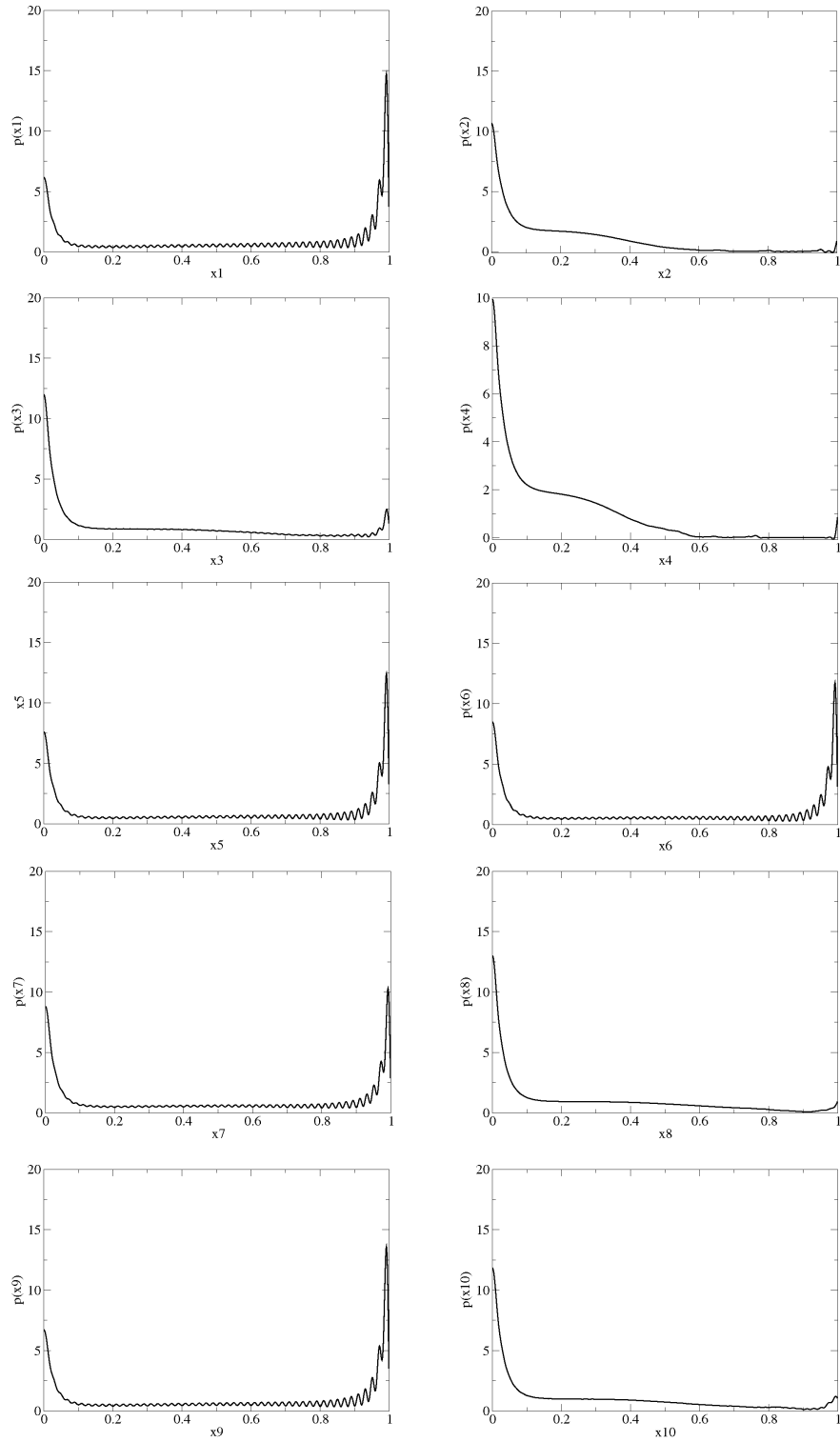


Figura 4.2: Estimación de las densidades de probabilidad para el problema de la mochila de diez variables con parámetros para el SFPL de $L = 100$ y $D = 0.5$.

4.3 SFPL: UN ENFOQUE BICRITERIO

Dado que se tiene la necesidad de encontrar una forma de como manejar las restricciones en problemas de optimización, nos encontramos en la literatura con un enfoque dado por Osyczka et al. [27] para transformar el problema original con restricciones de un simple criterio a un problema de optimización bicriterio en cual la primera función objetivo es la suma de las restricciones que son violadas y la segunda función objetivo es la función de costos que debe ser minimizada.

La siguiente formulación es un problema de programación no lineal para optimización de un simple criterio.

$$\begin{aligned}
 & \text{mín} \quad f(x) \\
 & \text{s. a.} \quad h_i(x) = 0 \quad i = 1, \dots, I \\
 & \quad \quad g_r(x) \leq 0 \quad r = 1, \dots, R \\
 & \quad \quad L_{1,n} \leq x_n \leq L_{2,n} \quad n = 1, \dots, N
 \end{aligned} \tag{4.6}$$

donde $h_i(x)$ son todas las restricciones de igualdad y $g_r(x)$ son todas las restricciones de desigualdad.

En el trabajo de Osyczka et al.[27] se propone la siguiente transformación del modelo original de un simple criterio a un modelo bicriterio,

$$\begin{aligned}
 & \text{mín}(f_1(x), f_2(x)) \\
 & f_1(x) = \sum_{i=1}^I |h_i| + \sum_{r=1}^R S(g_r)g_r \\
 & f_2(x) = f(x) \\
 & \text{sujeto a} \quad L_{1,n} \leq x_n \leq L_{2,n}
 \end{aligned} \tag{4.7}$$

donde $S(g)$ es una función de paso dada por,

$$S(g) = \begin{cases} 0 & \text{si } g \leq 0 \\ 1 & \text{si } g > 0 \end{cases}$$

Por construcción, el mínimo de $f_1(x)$ es $f_1(x) = 0$ y se localiza donde las restricciones son satisfechas. La función $f(x)$ es el objetivo original.

Inspirado en la formulación anterior, se propone separar el problema original en dos nuevos problemas independientes. Para el primer problema se propone estimar una densidad con la función de costos y en el segundo es estimar otra densidad con la función de restricciones. La formulación siguiente se propone para estimar las densidades de cada problema a través del SFPL.

$$\begin{aligned} & \text{mín} && f_c(x) \\ & \text{sujeta a} && L_{1,n} \leq x_n \leq L_{2,n} \end{aligned} \tag{4.8}$$

$$\begin{aligned} & \text{mín} && f_r(x) = \sum_{i=1}^I |h_i| + \sum_{r=1}^R S(g_r)g_r \\ & \text{sujeta a} && L_{1,n} \leq x_n \leq L_{2,n} \end{aligned} \tag{4.9}$$

donde $f_c(x)$ es la función objetivo de costos y $f_r(x)$ es la función objetivo de las restricciones.

El SFPL es un mecanismo de diversificación en donde a través de estimar densidades se puede obtener una población de puntos los cuales son muy consistentes con las propiedades óptimas globales, mismo que pueden ser aprovechados para el enfoque bicriterio dado por Osyczka et al. [27]. Para esto proponemos que se obtenga un conjunto de puntos de la densidad de la $p(f_c(x))$ y otro conjunto de puntos de la densidad de $p(f_r(x))$. Con estos dos conjuntos de puntos se puede formar otro conjunto de puntos con un espacio completo de soluciones para ambos

criterios el de $f_c(x)$ y $f_r(x)$. Asimismo este espacio completo de soluciones puede servir para encontrar una parte de la frontera pareto para el enfoque bicriterio, donde se minimizan ambos objetivos.

Para propósitos de este trabajo no es el de resolver el problema bicriterio, si no que solo se utiliza la formulación dada por Osyczka et al. [27] para tener una formulación similar para ser usada por el SFPL.

4.4 SFPL: UNA DENSIDAD CONJUNTA

Para este enfoque se utiliza la densidad estimada por cada uno de los objetivos dados en la formulación (4.8) para $f_c(x)$ y formulación (4.9) para $f_r(x)$ en la cual se obtiene una nueva densidad al multiplicar esas densidades independientes. A esta densidad se le llama densidad conjunta $p(f_c(x))p(f_r(x))$. Dicha densidad refleja las regiones que tienen las configuraciones óptimas de la solución. Estas regiones se encuentran en los puntos donde se satisfacen las restricciones y donde la función de costos tiene el mejor valor objetivo para el problema original con formulación (4.6).

Para ilustrar el desempeño de este enfoque propuesto, se usan dos funciones discutidas por Toscano Pulido et al. [36].

FUNCIONES DE PRUEBA:

Estas funciones de prueba son escogidas por contener características que son representativas de ser consideradas como «difíciles» en problemas de optimización global. Sus expresiones son dadas por las siguientes,

P.1:

$$\begin{aligned} \text{mín} \quad & x_1^2 + (x_2 - 1)^2 \\ \text{sujeta a} \quad & x_2 - x_1^2 = 0, \\ & -1 \leq x_1 \leq 1, \quad -1 \leq x_2 \leq 1, \end{aligned}$$

Solución: $(\pm 1/\sqrt{2}, 1/2)$.

P.2:

$$\begin{aligned} \min \quad & e^{x_1 x_2 x_3 x_4 x_5} \\ \text{sujeta a} \quad & x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0, \\ & x_2 x_3 - 5 x_4 x_5 = 0, \\ & x_1^3 + x_2^3 + 1 = 0, \\ & -2.3 \leq x_i \leq 2.3 \quad (i = 1, 2), \\ & -3.2 \leq x_i \leq 3.2 \quad (i = 3, 4, 5), \end{aligned}$$

Solución: $(-1.717143, 1.595709, 1.827247, -0.7636413, -0.763645)$.

Para el problema de prueba **P.1** se tiene una función cuadrática de dos variables. El problema de prueba **P.2** contiene diferentes tipos de restricciones combinadas como lineal, no lineal, igualdad y desigualdad en las que se tienen espacios de regiones factibles de solución muy pequeños o incluso disjuntos [36].

Dado el enfoque descrito en este trabajo se procede a utilizar el algoritmo SFPL. Como resultado obtenemos la Figura 4.3, que muestra las densidades asociadas con la función objetivo de costos y la Figura 4.4 que muestra las densidades estimadas asociadas con la función de restricciones para el problema de prueba **P.1**. Por último, en la Figura 4.5 se representa el producto $p(x_n|f_c)p(x_n|f_r)$ de las densidades aprendidas por SFPL para los objetivos $f_c(x)$ y $f_r(x)$ en el problema de prueba **P.1**. Es claro ver que la densidad conjunta tendrá valores grandes en las regiones donde se satisfacen las restricciones y mejora la función de costos. Se puede ver que la densidad conjunta describe perfectamente las regiones donde se encuentra la solución óptima global con alta probabilidad.

El enfoque propuesto de esta forma para el problema de prueba **P.1** obtiene las densidades de probabilidad asociadas a las configuraciones óptimas de solución del problema original. Las densidades se encuentran en regiones de solución consistentes con la solución óptima conocida.

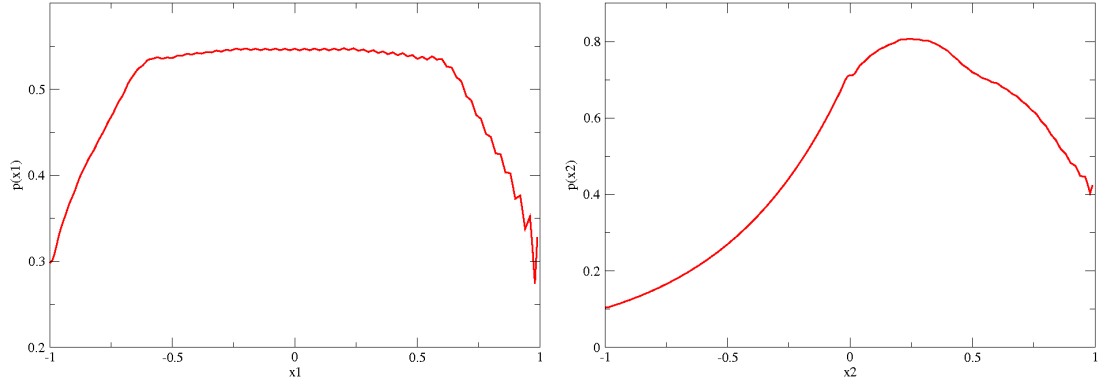


Figura 4.3: Estimación de las densidades asociadas a la función objetivo de costos $p(x_n|f_c)$ para el problema de prueba **P.1**, con parámetros de $L = 100$ y $D = 0.5$ para el SFPL.

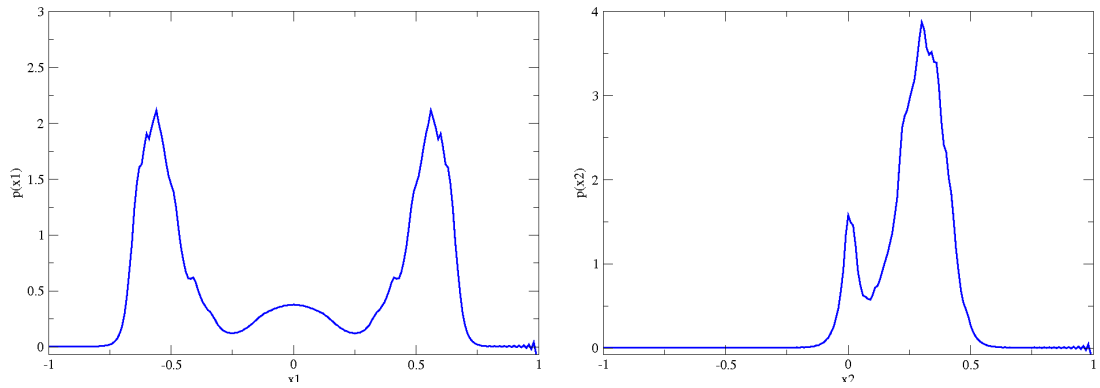


Figura 4.4: Estimación de las densidades asociadas a la función de restricciones $p(x_n|f_r)$ para el problema de prueba **P.1**, con parámetros $L = 200$ y $D = 0.03$ para el SFPL.

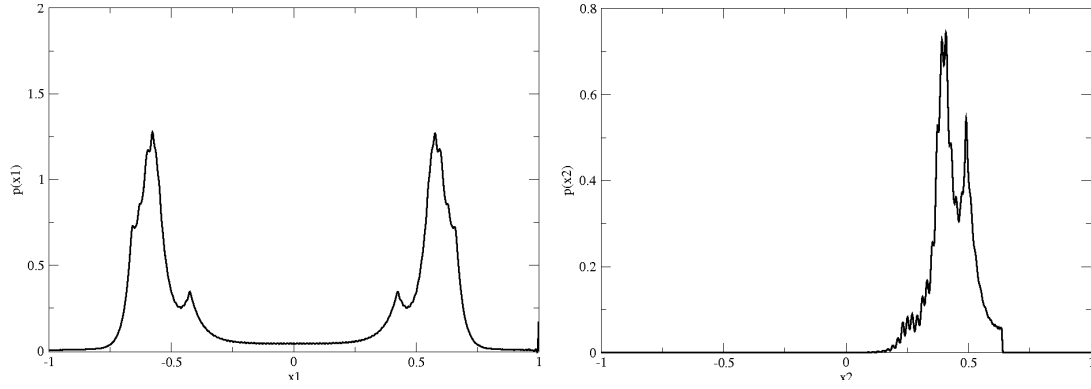


Figura 4.5: La densidad conjunta $p(x_n|f_c)p(x_n|f_r)$ obtenida para el problema de prueba **P.1**. La región de alta probabilidad de las densidades corresponden al óptimo global conocido.

El algoritmo SFPL realizó una iteración para estimar la densidad asociada a la función de costos, por el hecho de que la función es separable como se discute por Berrones [3], y 10 iteraciones para estimar la densidad asociada a la función de restricciones. Asimismo usando las formulas dadas por Berrones [3] se calculó el esfuerzo computacional medido con el número de evaluaciones de la función objetivo. Para la función de costo y de restricciones se tienen 396 y 7960 evaluaciones, respectivamente. Otros algoritmos heurísticos evolutivos, como optimización por enjambre de partículas (PSO por sus siglas en inglés, *Particle Swarm Optimization*), requiere de 340,000 evaluaciones a la función objetivo para el mismo problema de prueba **P.1** (discutido por Toscano Pulido et al. [36]). Con esto se tiene que el algoritmo SFPL requiere de pocas evaluaciones a la función objetivo en comparación con el PSO, por lo que esto implica un esfuerzo computacional menor.

4.5 SFPL: DENSIDAD PARA LA FUNCIÓN DE RESTRICCIONES

Las densidades estimadas por el algoritmo SFPL pueden servir como un mecanismo de diversificación en algoritmos heurísticos y exactos. Dichos algoritmos re-

quieren de puntos iniciales factibles y de buena calidad que ayuden a los mismos a converger a una solución óptima o aproximada con un esfuerzo computacional menor.

El SFPL nos ayuda a estimar densidades con la función de costo o la de restricciones para aprender las regiones en donde se puede tomar una población de puntos. Esto es posible con el método para muestrear la densidad de máxima probabilidad. Esta población de puntos tomados de la densidad satisfacen a la misma función, por el hecho de que fue un aprendizaje con esa función. Con este fin se usa el SFPL para el aprendizaje de la función de restricciones. Se estima la densidad para dicha restricción y se obtiene una población de puntos. Esa población de puntos satisfacen el conjunto de restricciones, teniendo con esto puntos factibles y de buena calidad ya que se encuentran en la vecindad del óptimo global con alta probabilidad. Con esto se espera que los algoritmos exactos o heurísticos encuentren soluciones óptimas o muy aproximadas.

A continuación se presenta una comparación de la densidad uniforme y una densidad estimada por el SFPL. La comparación consiste en tomar puntos de ambas densidades y evaluarlos en la función de restricciones para los problemas de prueba **P.1** y **P.2**. Con esto se determina la tasa de puntos encontrados que satisfacen a las restricciones con una tolerancia de ϵ dada. En los cuales si la evaluación $f_r(x) < \epsilon$, estos puntos se consideran factibles. La tasa se determina de los puntos factibles encontrados entre el número total de puntos evaluados.

En la Tabla 4.1 se presentan los resultados de la comparación entre la densidad uniforme y la densidad SFPL. Para el problema de prueba **P.1** con 100 iteraciones del algoritmo SFPL, se evalúan 100 puntos tomados de la densidad uniforme y 100 puntos tomados de la densidad SFPL con tolerancia de $\epsilon = 0.01$, donde se puede notar que la densidad SFPL tiene una tasa mayor con respecto a la densidad uniforme dado que los puntos tomados de ella contiene 29 puntos factibles y la densidad uniforme solo 2 puntos factibles. Para el mismo problema pero con 500

	Iteraciones	Densidad SFPL	Densidad uniforme
P.1	100	0.29	0.02
	500	0.32	0.06
P.2	100	0.61	0.00
	500	0.556	0.00

Tabla 4.1: Resultados de la comparación de la densidad uniforme y una densidad estimada por el SFPL para los problemas de prueba **P.1** y **P.2** con tolerancia de $\epsilon = 0.01$ y $\epsilon = 0.1$ respectivamente.

iteraciones se puede notar que la tasa de la densidad SFPL es mayor con respecto a la densidad uniforme, por lo que de igual forma la densidad SFPL obtiene un mayor número de puntos factibles. Para el problema de prueba **P.2** con 100 iteraciones del algoritmo SFPL se evalúan 100 puntos tomados de la densidad uniforme y 100 puntos tomados de la densidad SFPL con tolerancia de $\epsilon = 0.1$, donde se puede notar que para el conjunto de restricciones de ese problema no encuentra puntos factibles con la densidad uniforme, mientras que tomar puntos de la densidad SFPL 61 puntos son factibles. Para el mismo problema pero con 500 iteraciones se tiene un alto número de puntos factibles con la densidad SFPL, mientras que la densidad uniforme no obtiene ningún punto factible. Nótese que para los problemas de prueba se utilizó un ϵ diferente debido a la dificultad de satisfacer a las restricciones como se discutió en secciones anteriores.

Ahora se realiza una comparación de generar un conjunto de soluciones dadas por la densidad del SFPL con respecto a generar un mayor número de soluciones aleatorias. El tamaño del conjunto de soluciones generadas aleatoriamente (discutido por Toscano Pulido et al. [36]) es de 1,000,000 y el conjunto de soluciones generadas por la densidad del SFPL es de 10,000. Con esto se procede a usar la misma idea discutida anteriormente, la cual consiste en tomar puntos de la densidad SFPL y después evaluarlos en la función de restricciones para los problemas de prueba **P.1** y **P.2**. Con éstos se determina la tasa de puntos encontrados que satisfacen a las

restricciones con la misma tolerancia de $\epsilon = 0.01$ para ambos problemas.

	Algoritmo SFPL	Soluciones aleatorias
P.1	0.3151	0.0973
P.2	0.5723	0.0000

Tabla 4.2: Se muestra la tasa de puntos que satisfacen a las restricciones para 10,000 soluciones del SFPL y 1,000,000 de soluciones aleatorias generadas para los problemas **P.1** y **P.2** con tolerancia de $\epsilon = 0.01$ en ambos problemas.

La Tabla 4.2 presenta la comparación de los resultados obtenidos por Toscano Pulido et al. [36] y los resultados obtenidos por el desempeño del SFPL. Se puede ver que al generar 1,000,000 de soluciones aleatorias para el problema de prueba **P.1**, la tasa de soluciones factibles encontradas es muy pequeña. Por otro lado, el algoritmo SFPL obtuvo una tasa de soluciones factibles mayor en comparación a generar las soluciones de forma aleatoria. Nótese que el SFPL solo generó 10,000 soluciones por lo que se espera que para 1,000,000 esta tasa sea mucho mayor. En el caso del problema de prueba **P.2** la tasa de soluciones factibles encontradas por el SFPL es relativamente alta, mientras que las soluciones aleatorias generadas no contienen soluciones factibles.

Con lo anterior, los resultados indican que el SFPL brinda una sustancial mejora con respecto a generar soluciones aleatorias de una densidad uniforme en ambos casos de prueba. Se puede ver que las densidades estimadas por el algoritmo SFPL sirven como mecanismo de diversificación en algoritmos heurísticos y algoritmos exactos que describen regiones donde se satisfacen las restricciones con alta probabilidad. Estos algoritmos heurísticos y exactos requieren de puntos iniciales factibles y de buena calidad que ayuden a la convergencia de una solución óptima o muy aproximada con un esfuerzo computacional menor. El algoritmo SFPL brinda una población de puntos que pueden ser aprovechados en otros algoritmos de optimización.

4.6 LOS ALGORITMOS GENÉTICOS

4.6.1 ANTECEDENTES

Los algoritmos genéticos son métodos adaptativos que pueden usarse para intentar resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin [6]. Por imitación de este proceso, los algoritmos genéticos son capaces de ir creando soluciones para problemas del mundo real.

Los principios básicos de los algoritmos genéticos fueron establecidos por Holland [17], y se encuentran bien descritos Goldberg [15] y Michalewicz [25].

En la naturaleza, los individuos de una población compiten entre sí en la búsqueda de recursos tales como comida, agua y refugio. Incluso los miembros de una misma especie compiten a menudo en la búsqueda de un compañero. Aquellos individuos que tienen más éxito en sobrevivir y en atraer compañeros tienen mayor probabilidad de generar un gran número de descendientes. Por el contrario individuos poco dotados producirán un menor número de descendientes. Esto significa que los genes de los individuos mejor adaptados se propagarán en sucesivas generaciones hacia un número creciente de individuos. La combinación de buenas características provenientes de diferentes ancestros, puede a veces producir descendientes mejores llamados superindividuos, cuya adaptación es mucho mayor que la de cualquiera de sus ancestros. De esta manera, las especies evolucionan logrando unas características cada vez mejor adaptadas al entorno en el que viven.

Los algoritmos genéticos usan una analogía directa con el comportamiento natural. Trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema dado. A cada individuo se le asigna un valor, relacionado con la bondad de dicha solución. En la naturaleza esto equivaldría al gra-

do de efectividad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma. Este cruce producirá nuevos individuos los cuales comparten algunas de las características de sus padres. Cuanto menor sea la adaptación de un individuo, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones. De esta manera se produce una nueva población de posibles soluciones, la cual reemplaza a la anterior y verifica la interesante propiedad de que contiene una mayor proporción de buenas características en comparación con la población anterior. Así a lo largo de las generaciones las buenas características se propagan a través de la población. Favoreciendo el cruce de los individuos mejor adaptados, van siendo exploradas las áreas más prometedoras del espacio de búsqueda. En la Figura 4.6 se muestra el esquema general de un algoritmo genético. Si el algoritmo genético está bien diseñado, la población convergerá hacia una solución del problema muy aproximada.



Figura 4.6: Diagrama general de un algoritmo genético.

El poder de los algoritmos genéticos proviene del hecho de que se trata de una técnica robusta, y pueden tratar con éxito una gran variedad de problemas prove-

nientes de diferentes áreas, incluyendo aquellos en los que otros métodos encuentran dificultades. Si bien no se garantiza que el algoritmo genético encuentre la solución óptima del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable, en un tiempo competitivo con el resto de algoritmos de optimización. En el caso de que existan técnicas especializadas para resolver un determinado problema, lo más probable es que superen al algoritmo genético, tanto en rapidez como en eficacia. El gran campo de aplicación de los algoritmos genéticos se relaciona con aquellos problemas para los cuales no existen técnicas especializadas. Incluso en el caso en que dichas técnicas existan, y funcionen bien, pueden efectuarse mejoras de las mismas hibridándolas con los algoritmos genéticos.

4.6.2 GENERALIDADES

Los individuos o posibles soluciones del problema, pueden representarse como un conjunto de valores de las variables que denominan genes, los cuales agrupados forman un vector de valores que son referidos como cromosoma. El alfabeto utilizado para representar los individuos no debe necesariamente estar constituido por el $[0, 1]$, buena parte de la teoría en la que se fundamentan los algoritmos genéticos utiliza dicho alfabeto.

En términos biológicos, el conjunto de valores de las variables representando un cromosoma particular se denomina fenotipo. El fenotipo contiene la información requerida para construir un organismo, el cual se refiere como genotipo. Los mismos términos se utilizan en el campo de los algoritmos genéticos.

La adaptación al problema de un individuo depende de la evaluación del genotipo. Esto último puede inferirse a partir del fenotipo, es decir puede ser calculada a partir del cromosoma, usando la función de evaluación. La función de adaptación debe ser diseñada para cada problema de manera específica. Dado un cromosoma particular, la función de adaptación le asigna un valor, que se supone refleja el nivel de adaptación al problema del individuo representado por el cromosoma.

Durante la fase reproductiva se seleccionan los individuos de la población para cruzarse y producir descendientes, que constituirán, una vez mutados, la siguiente generación de individuos. La selección de padres se efectúa al azar usando un procedimiento que favorezca a los individuos mejor adaptados, ya que a cada individuo se le asigna una probabilidad de ser seleccionado que es proporcional a su función de adaptación. Este procedimiento se dice que está basado en la ruleta sesgada. Según dicho esquema, los individuos bien adaptados se escogerán probablemente varias veces por generación, mientras que los pobremente adaptados al problema, no se escogerán más que de vez en cuando. Una vez seleccionados dos padres, sus cromosomas se combinan, utilizando habitualmente los operadores de cruce y mutación.

El operador de cruce, escoge dos padres seleccionados y corta en dos segmentos a los cromosomas en una posición escogida al azar, para producir dos segmentos iniciales y dos segmentos finales. Después se intercambian los segmentos finales, produciéndose dos nuevos cromosomas completos. Ambos descendientes heredan genes de cada uno de los padres. Este operador se conoce como operador de cruce basado en un punto. En la Figura 4.7 se muestra un ejemplo para el operador de cruce basado en un punto. Habitualmente el operador de cruce no se aplica a todos los pares de individuos que han sido seleccionados para emparejarse, sino que se aplica de manera aleatoria. En el caso en que el operador de cruce no se aplique, la descendencia se obtiene simplemente duplicando los padres.

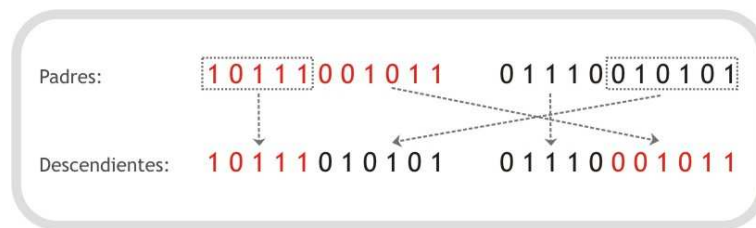


Figura 4.7: Operador de cruce para el algoritmo genético basado en un punto de corte.

El operador de mutación se aplica a cada hijo de manera individual y consiste en la alteración aleatoria con probabilidad pequeña de cada gen componente del

cromosoma. En principio se puede pensar que el operador de cruce es más importante que el operador de mutación ya que proporciona una exploración rápida del espacio de búsqueda. El operador de mutación asegura que ningún punto del espacio de búsqueda tenga probabilidad cero de ser examinado, y esto es de gran importancia para asegurar la convergencia de los algoritmos genéticos.

La definición de convergencia fue introducida en este campo por De Jong [7]. Si el algoritmo genético ha sido correctamente implementado, la población evolucionará a lo largo de las generaciones sucesivas de tal manera que la adaptación media de todos los individuos de la población, así como la adaptación del mejor individuo irán convergiendo hacia el óptimo global. El concepto de convergencia está relacionado con la progresión hacia la uniformidad, un gen ha convergido cuando al menos el 95 % de los individuos de la población comparten el mismo valor para dicho gen. Se dice que la población converge cuando todos los genes han convergido.

Se presenta a continuación el pseudocódigo para un AGS (por sus siglas en español, *Algoritmo Genético Simple*) que es propuesto por Michalewicz [25]. Los métodos utilizados en el pseudocódigo como `Funcion_de_seleccion()`, `Opera_Cruce()`, `Opera_Mutar()` se explican más adelante.

Algoritmo 4.1 *AGS(Prob_muta, Prob_cruce, M)*

Input: *Prob_muta*: probabilidad de mutación del gen; *Prob_cruce*: probabilidad de cruce del individuo; *M*: número de iteraciones del algoritmo.

Output: El mejor individuo de la población.

Contador \leftarrow 0 ;

Poblacion \leftarrow Generar_pobla_inicial_aleat();

Poblacion \leftarrow Func_elitismo(*Poblacion*);

while (*Contador* < *M*) **do**

Pob_Selec \leftarrow Función_de_Seleccion(*Poblacion*) ;

Indiv_a_mutar \leftarrow Opera_Cruce(*Prob_cruce*, *Pob_Selec*) ;

Pob_muta \leftarrow Opera_Mutar(*Prob_muta*, *Poblacion* \cup *Indiv_a_mutar*);

```

Pob_muta ← Func_elitismo(Pob_muta) ;
Poblacion ← Borrar_peores_indiv(Pob_muta) ;
Contador ++ ;
end while
return Obten_mejor(Poblacion) ;

```

Aquí Func_elitismo() reemplaza al peor de los individuos por una réplica del mejor de los mismos. El método Borrar_peores_indiv() elimina los peores individuos de la población dejando el mismo tamaño de la población inicial para ser la nueva población en la siguiente iteración. Por último, el método Obten_mejor() toma el mejor individuo de la población al final de las iteraciones para ser reportado como solución.

4.7 HIBRIDACIÓN DE SFPL CON ALGORITMOS GENÉTICOS

Dada la naturaleza del algoritmo SFPL como mecanismo de diversificación, las soluciones obtenidas a través de estimar las densidades por el algoritmo SFPL pueden ser ligadas a algún mecanismo de adaptación. En primera instancia, se desarrolla un algoritmo genético simple con algunas modificaciones. La adaptación del algoritmo genético está dada con el fin de contraer a la población de soluciones a una región donde toman su mejor valor de la función. La población de soluciones se toman de las densidades estimadas por el SFPL en regiones de máxima probabilidad. El hecho de usar el potencial del SFPL con los algoritmos genéticos tiene como resultante un algoritmo híbrido «SFPL-AG» para tratar a problemas de optimización. Los detalles del algoritmo híbrido que se usa en este trabajo se describen a continuación.

4.7.1 POBLACIÓN

TAMAÑO DE POBLACIÓN

Una cuestión que uno puede plantearse es la relacionada con el tamaño idóneo de la población. Parece intuitivo que las poblaciones pequeñas corren el riesgo de no cubrir adecuadamente el espacio de búsqueda, mientras que el trabajar con poblaciones de gran tamaño puede acarrear problemas relacionados con el excesivo costo computacional. Basados en evidencia empírica Goldberg [15], sugiere que un tamaño de población de 30 es suficiente para atacar con éxito a los problemas considerados en su trabajo. Inspirados en ese tamaño de población se realizó la experimentación del parámetro con diferentes tamaños. Se observó que para tamaños grandes de población no se obtenían mejores resultados con respecto a un tamaño de población pequeño no menor de 30. De esta forma se determinó que una población de 30 individuos es suficiente para el resto de los experimentos.

POBLACIÓN INICIAL

Habitualmente la población para los algoritmos genéticos es generada al azar. Nos podríamos preguntar que es lo que sucedería si los individuos de la población inicial se obtuviesen como resultado de alguna técnica heurística o de optimización local. En los pocos trabajos que existen sobre este aspecto, se constata que esta inicialización no aleatoria de la población inicial, puede acelerar la convergencia del algoritmo genético. Sin embargo en algunos casos la desventaja resulta ser la prematura convergencia del algoritmo, queriendo indicar con ésto la convergencia hacia óptimos locales.

Por lo anterior mencionado, dado que el SFPL estima densidades en regiones donde se localizan buenas soluciones con alta probabilidad, la población puede ser tomadas de ellas para que la solución sea adaptada en esa región donde está contenida la solución óptima global. Con esto tendríamos una convergencia rápida hacia una

solución óptima local en la región de alta probabilidad.

Mediante el muestreo de la densidad, como se ve en la Figura 4.8, se pueden obtener puntos en la vecindad del óptimo global. La cercanía de los puntos en la región de máxima probabilidad depende de que tan ajustada es la densidad. Para generar la población de puntos iniciales en el algoritmo genético, se procede a tomar el punto máximo en las densidades para obtener el primer individuo, y posteriormente muestrear a las densidades para obtener el resto de los individuos de la población. Con esto se tiene una población de puntos donde está contenido el individuo máximo de la densidad y los individuos de la población diversa en esa región de la densidad.

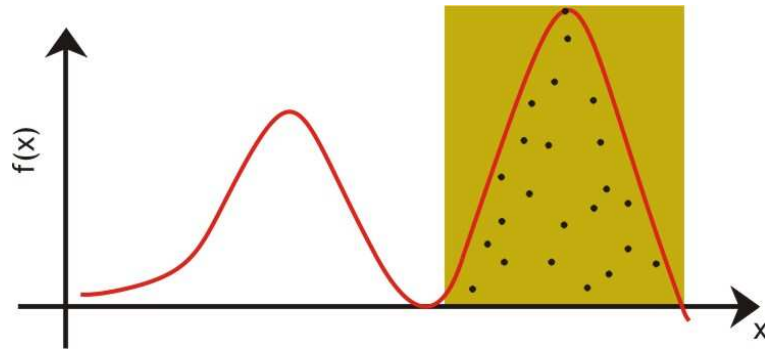


Figura 4.8: Densidad estimada por el SFPL que es muestreada en la región de alta probabilidad

4.7.2 FUNCIÓN OBJETIVO

En los casos en donde existe una función objetivo sin ser sujeta a restricciones, los algoritmos genéticos trabajan con la función objetivo en su forma simple sobre un espacio de búsqueda definido. En los algoritmos evolutivos y los genéticos, la función objetivo es llamada también como la función de evaluación o adaptación. Por otro lado, para el planteamiento en el que los individuos están sometidos a restricciones, se han propuesto varias soluciones. La primera sería la denominada técnica absolutista, en la que aquellos individuos que no satisfacen las restricciones, no son considerados como tales, y se siguen efectuando cruces y mutaciones hasta obtener individuos válidos, o bien a dichos individuos se les asigna una función

objetivo igual a cero. Otra posibilidad consiste en reconstruir aquellos individuos que no satisfacen las restricciones. Dicha reconstrucción suele llevarse a cabo por medio de un nuevo operador que se acostumbra denominar como reparador.

Otro enfoque está basado en la penalización de la función objetivo. La idea general consiste en dividir la función objetivo del individuo por un factor de penalización que guarda relación con las restricciones que dicho individuo viola. Dicha cantidad puede simplemente tener en cuenta el número de restricciones violadas ó bien el denominado costo esperado de reconstrucción, es decir el costo asociado a la conversión de dicho individuo en otro que no viole ninguna restricción.

En nuestro caso para los problemas sin restricciones se usa la función objetivo de la ecuación (4.8). En el caso de los problemas con restricciones se propone tomar la misma función objetivo de la ecuación (4.8) pero con una población de la densidad conjunta estimada como se muestra en la Figura 4.5. Asimismo se propone un nuevo enfoque para el manejo de las restricciones en los algoritmos genéticos que más adelante se describe.

4.7.3 SELECCIÓN

Para generar la nueva población de individuos padre, la función de selección de padres más utilizada es la denominada función de selección proporcional a la función objetivo, en la cual cada individuo tiene una probabilidad de ser seleccionado como padre que es proporcional al valor de su función objetivo [15]. En nuestro caso, se utiliza una forma simple de selección de la población para el cruce, la que consiste en seleccionar los individuos al azar del conjunto de individuos de la población inicial. Esto es, generar numeros aleatorios enteros entre 1 y el tamaño de la población. Con esto se tiene la nueva población de individuos padre que son tomados al azar de la población inicial para ser cruzado. Esta nueva población de individuos padre es del mismo tamaño que la población inicial.

4.7.4 CRUCE

El operador de cruce esencialmente combina las características de dos individuos padres que forman a dos individuos hijos similares por correspondientes intercambios de segmentos de los padres. La intuición detras de la aplicación del cruce es el intercambio de información entre diferentes soluciones potenciales para generar una con mejores características.

Existen diversos operadores de cruce para combinar a los individuos con valores tipo binario o continuo. Para nuestro enfoque SFPL se requiere un operador que trabaje con valores continuos. Michalewicz [25] propone el operador de cruce aritmético para individuos con valores continuos. En [25] se discute este operador, donde bajo experimentación realizada se determinó que es el que tiene mejor desempeño entre los operadores discutidos para valores continuos. Por tal motivo en el presente trabajo se usa dicho operador de cruce aritmético. En la Figura 4.9 se muestra un ejemplo para el operador de cruce aritmético.

CRUCE ARITMÉTICO: Este operador es definido como una combinación lineal de dos individuos:

Si los padres x_1 y x_2 son cruzados, los hijos resultantes son $x'_1 = a \cdot x_1 + (1-a) \cdot x_2$ y $x'_2 = a \cdot x_2 + (1-a) \cdot x_1$. Este operador usa un valor aleatorio $a \in (0, 1)$.

Para seleccionar que individuos serán cruzados se tiene una probabilidad de cruce p_c asociada y se procede a realizar lo siguiente,

- Generar un número aleatorio uniforme $r \in (0, 1)$.
- Si $r < p_c$, se selecciona el individuo para el cruce.

Si el número de padres a cruzar es impar se puede elegir un nuevo individuo o también se puede remover uno aleatoriamente. Si el número de padres es par se deja la lista de individuos igual.

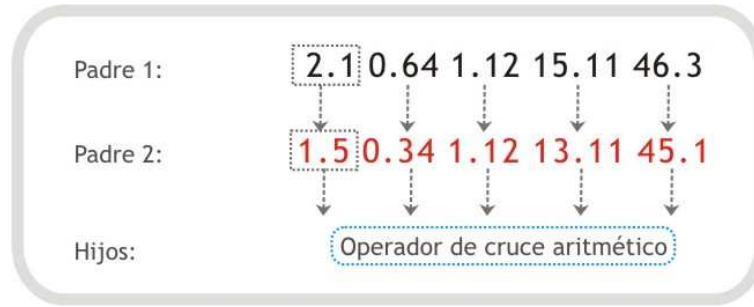


Figura 4.9: Operador de cruce aritmético para el algoritmo genético.

4.7.5 MUTACIÓN

La mutación arbitrariamente altera uno o más genes de un individuo seleccionado, por un cambio aleatorio con una probabilidad igual a la tasa de mutación. La intuición detrás del operador de mutación es la introducción de alguna variabilidad extra en la población.

Para seleccionar que individuos serán cruzados se tiene una probabilidad de cruce p_c asociada. Cada gen de todos los individuos de la población tiene igual oportunidad de ser mutado. El valor que toma el gen a mutar se intercambia por un valor de la densidad del SFPL muestreada. Este valor es remplazado en esa posición. Para esto se procede a realizar lo siguiente,

- Generar un número aleatorio uniforme $r \in (0, 1)$.
- Si $r < p_m$, se selecciona el gen para ser mutado.

La población que se toma en cuenta para realizar la mutación es la población de individuos iniciales y los individuos hijos provenientes del cruce.

4.7.6 PARÁMETROS

Diversos estudios para determinar los parámetros adecuados en un algoritmo genético se han realizado. En una serie de estudios paramétricos realizados por

Goldberg [15], sugiere que para un buen desempeño de un algoritmo genético, requiere de la elección de una alta probabilidad para el operador de cruce, una baja probabilidad para el operador de mutación que será inversamente proporcional al tamaño de la población, y un tamaño de población moderada.

Para los experimentos realizados en este trabajo para todos los problemas se usaron los siguientes parámetros recomendados,

- Probabilidad de mutación (p_m) = 0.0333
- Probabilidad de cruce (p_c) = 0.6
- Tamaño de población ($popsize$) = 30

4.7.7 ALGORITMO DE SFPL-AGI

En este trabajo se propone el siguiente algoritmo híbrido SFPL-AGI. Para este enfoque se propone una estimación de las densidades a través del algoritmo SFPL, hasta que las iteraciones del SFPL han terminado. Después se genera una población inicial de puntos, que está integrada por la población de puntos muestreados de la densidad estimada, así como el punto máximo de la misma densidad. Dicha población inicial se proveerá al algoritmo genético. Después se procede a la adaptación de la solución a través del algoritmo genético. La salida del algoritmo genético es el individuo que fue mejor adaptado a través de las generaciones. A continuación se describe los pasos del algoritmo híbrido SFPL-AGI.

Algoritmo 4.2 *SFPL-AGI*($Prob_muta$, $Prob_cruce$, M)

Input: $Prob_muta$: probabilidad de mutación del gen; $Prob_cruce$: probabilidad de cruce del individuo; M : número de iteraciones del algoritmo.

Output: El mejor individuo de la población.

$Contador \leftarrow 0$;

```

Poblacion  $\leftarrow$  SFPL();
Poblacion  $\leftarrow$  Func_elitismo(Poblacion);
while (Contador < M) do
    Pob_Selec  $\leftarrow$  Función_de_Seleccion(Poblacion) ;
    Indiv_a_mutar  $\leftarrow$  Opera_Cruce(Prob_cruce, Pob_Selec) ;
    Pob_muta  $\leftarrow$  Opera_Mutar(Prob_muta, Poblacion  $\cup$  Indiv_a_mutar);
    Pob_muta  $\leftarrow$  Func_elitismo(Pob_muta) ;
    Poblacion  $\leftarrow$  Borrar_peores_indiv(Pob_muta) ;
    Contador++;
end while
return Obten_mejor(Poblacion) ;

```

A continuación se presentan los detalles de los resultados dados por la ejecución del algoritmo SFPL-AGI para el problema de Booth anteriormente discutido.

```

***SFPL***

Parámetros SFPL:
NVAR: 2, Seed: 337, M: 100, L: 100, D: 10.0;
LMIN1[-10], LMIN2[-10], LMAX1[10], LMAX2[10];
Estimación de la densidad completada...
Máximo de la densidad (1): 1.0999999999999996
Máximo de la densidad (2): 3.08
Valor objetivo: 0.145999999999999952

***Algoritmo genético***

No. de Generaciones: 200
Mejor individuo (1): 1.0001331021970348
Mejor individuo (2): 2.999881970707216
Valor objetivo: 3.255587858076561E-8

```

En la Figura 4.10 se muestra la comparación de la convergencia del AGS con respecto al algoritmo SFPL-AGI. Se observa que el algoritmo de SFPL-AGI ha

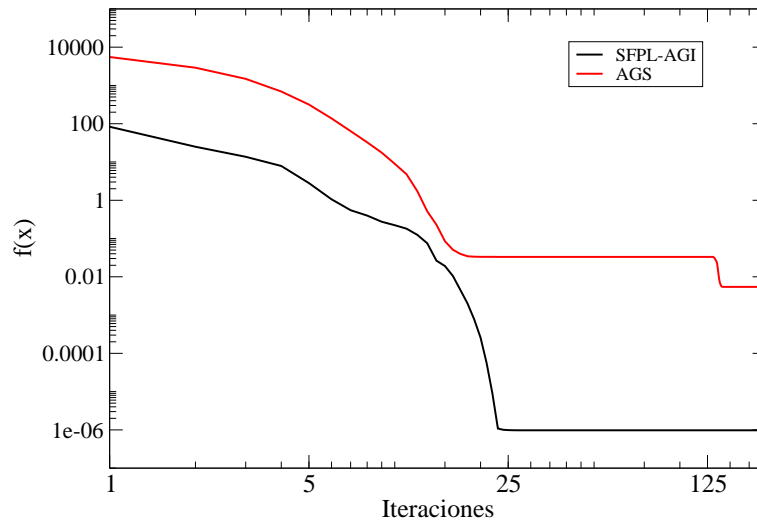


Figura 4.10: Comparación de la convergencia del AGS contra el algoritmo SFPL-AGI para el problema de Booth.

mejorado sustancialmente la eficiencia de los algoritmos genéticos. Esta mejora viene dada por el hecho de que el algoritmo SFPL-AGI requiere de menos iteraciones para encontrar la solución mucho más aproximada a la solución óptima. Esto se debe a que la población inicial está dada de las densidades estimadas por el SFPL en regiones de máxima probabilidad en vez de una población inicial aleatoria.

4.7.8 ALGORITMO DE SFPL-AGII

Este es un segundo enfoque de algoritmo híbrido. En este enfoque se propone la estimación de las densidades a través del algoritmo SFPL que iteración tras iteración brindará una población de puntos para el algoritmo genético. En la primera iteración de SFPL, se toma una población de puntos muestreados de la densidad estimada y se toma el punto máximo de la misma densidad. Dichos puntos son la población inicial para el algoritmo genético. Después se procede a adaptar la solución en la región donde tome su mejor valor objetivo a través del algoritmo genético. El algoritmo SFPL-AGII guarda la mejor solución adaptada que se encontró en esa iteración.

De la misma forma guarda todas las mejores soluciones adaptadas encontradas en cada iteración hasta que el algoritmo híbrido termina sus iteraciones. Al final del algoritmo SFPL-AGII, se reporta la mejor solución tras la evaluación de todas las soluciones encontradas en las M iteraciones. A continuación se describen los pasos del segundo algoritmo propuesto.

Algoritmo 4.3 *SFPL-AGI(Prob_muta, Prob_cruce, M)*

Input: *Prob_muta*: probabilidad de mutación del gen; *Prob_cruce*: probabilidad de cruce del individuo; *M*: número de iteraciones del algoritmo.

Output: El mejor individuo de la población.

```

Contador  $\leftarrow$  0 ;
for  $i = 1, \dots, M$  do
    Poblacion  $\leftarrow$  SFPL(i);
    Poblacion  $\leftarrow$  Func_elitismo(Poblacion);
    while (Contador < M) do
        Pob_Selec  $\leftarrow$  Función_de_Seleccion(Poblacion) ;
        Indiv_a_mutar  $\leftarrow$  Opera_Cruce(Prob_cruce, Pob_Selec) ;
        Pob_muta  $\leftarrow$  Opera_Mutar(Prob_muta, Poblacion  $\cup$  Indiv_a_mutar);
        Pob_muta  $\leftarrow$  Func_elitismo(Pob_muta) ;
        Poblacion  $\leftarrow$  Borrar_peores_indiv(Pob_muta) ;
        Contador++;
    end while
    Lista_Soluciones  $\leftarrow$  Obten_mejor(Poblacion) ;
end for
return Obten_mejor(Lista_Soluciones) ;

```

Los resultados mostrados en la Tabla 4.3 son para el mismo problema de Booth. Se puede ver que iteración tras iteración del algoritmo SFPL-AGII, se consigue tener una lista de soluciones que son bien adaptadas en cada iteración. Al final del algoritmo son evaluadas las soluciones según su valor de adaptación, o dicho de otra

forma, las de mejor valor objetivo y se toma como solución aquella que tenga el mejor valor adaptado.

Iteración	Individuos	Objetivo
5	(1.0001 , 2.9999)	2.0E-8
25	(0.9999 , 3.0000)	5.0E-8
35	(1.0003 , 2.9973)	3.042E-5
41	(0.9992 , 3.0004)	1.44E-6
57	(1.0003 , 2.9996)	2.9E-7
58	(0.9998 , 2.9999)	4.1E-7
69	(1.0061 , 2.9941)	7.218E-5
76	(0.9901 , 3.0060)	1.9485E-8
97	(0.9999 , 2.9999)	1.8E-7
100	(0.9979 , 3.0011)	9.62E-6

Tabla 4.3: Lista de las 10 mejores soluciones encontradas por el algoritmo híbrido SFPL-AGII para el problema de Booth tras 100 iteraciones.

A continuación se lleva a cabo una comparación entre los tres algoritmos (AGS, SFPL-AGI y SFPL-AGII). Para los dos primeros se utilizó 200 iteraciones y para el algoritmo SFPL-AGII se utilizó solo 100 iteraciones. En la Tabla 4.4 se muestran los resultados de esta comparación para el problema de Booth para los tres algoritmos. Se puede ver que en la ejecución del AGS se obtiene un valor objetivo de $f(x^*) = 1.8178188303693497^{-4}$. Por otro lado, para la ejecución del SFPL-AGI se obtiene un valor objetivo de $f(x^*) = 3.255587858076561^{-8}$. Aunque la diferencia de estos valores en términos absolutos es pequeña, la diferencia relativa es abismal siendo el SFPL-AGI cuatro órdenes de magnitud mejor que el AGS. Por último, el algoritmo SFPL-AGII obtiene una solución con valor objetivo de $f(x^*) = 6.699772591099562^{-20}$, con una diferencia relativa de doce órdenes de magnitud mejor con respecto al valor objetivo obtenido por SFPL-AGI. Con esto se puede observar que el algoritmo SFPL-AGII encuentra una solución más aproximada a la solución óptima conocida.

Método	x_1^{mejor}	x_2^{mejor}	$f(x^*)$
AGS	0.9918425000918	3.010047398355	1.8178188303693^{-4}
SFPL-AGI	1.0001331021970	2.999881970707	3.255587858076^{-8}
SFPL-AGII	0.9999999999403	2.999999999937	6.699772591099^{-20}

Tabla 4.4: Comparación de los resultados de los tres algoritmo (AGS, SFPL-AGI y SFPL-AGII) para el problema de Booth. Se realizó 200 iteraciones para los dos primeros algoritmos y 100 iteraciones para el tercer algoritmo.

El enfoque SFPL-AGII se puede utilizar para obtener una población de puntos de las densidades estimadas para la función de las restricciones, así como de las densidades estimadas para la función objetivo. Dicha población de puntos que se extraen de esas densidades son bien adaptadas por el algoritmo genético. Estas poblaciones de puntos pueden ser muy útiles para el enfoque bicriterio discutido anteriormente.

4.8 UN ENFOQUE DE MANEJO DE RESTRICCIONES EN ALGORITMOS GENÉTICOS

Como se ha visto anteriormente, el desempeño de los algoritmos híbridos propuestos han mejorado en gran medida con la ayuda del SFPL, dado que la población inicial de puntos se encuentran en regiones de máxima probabilidad, mismas que con alta probabilidad contienen a la solución óptima global. Dado que se ha observado el desempeño del SFPL para estimar densidades de la función objetivo como también de la función de restricciones, se pueden obtener ambas densidades para las funciones y trabajarlas de manera independiente. El método consiste en estimar una densidad de la función de restricciones, misma que dará como resultado una población de puntos que satisfacen a las restricciones como fue discutido anteriormente. A estas soluciones también se les conoce como soluciones factibles. Después esa población de puntos factibles se proveen al algoritmo genético para adaptar la solución ahora

satisfaciendo a la función objetivo. Con esto se pretende tener una solución que satisfice a las restricciones y que tiene el mejor valor según la función objetivo. Con esto se pretende tener un enfoque del SFPL como mecanismo para el manejo de restricciones¹ en los algoritmos genéticos para intentar encontrar una solución al problema de optimización.

El procedimiento para trabajar a las restricciones es el Algoritmo 4.7.7. Como se mencionó anteriormente, se pretende que el SFPL trabaje con las restricciones como se describe en la ecuación (4.9) y el algoritmo genético trabaje con la función objetivo de la ecuación (4.8). Los resultados de este enfoque para el problema de prueba **P.1** son presentados a continuación.

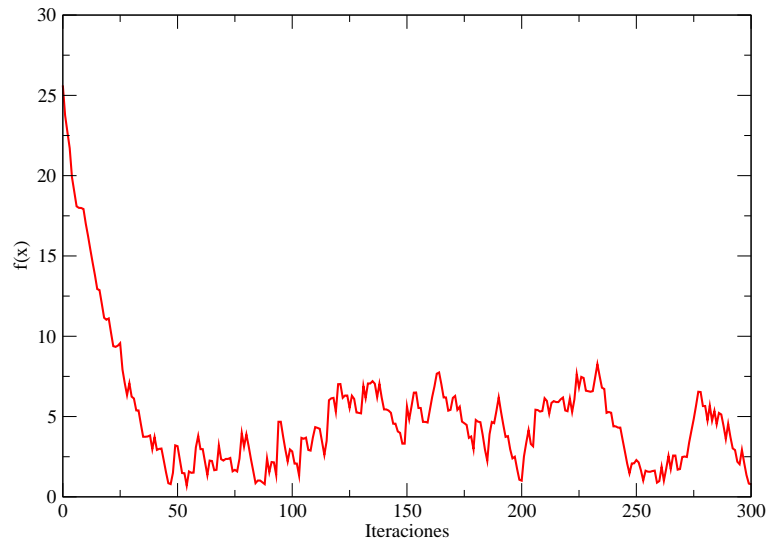


Figura 4.11: Convergencia del algoritmo SFPL-AGI según su valor objetivo para el problema de prueba P.1.

-

Para el enfoque de manejo de restricciones que es propuesto en este trabajo se tiene que tras manejar el SFPL con la función de restricciones se encuentra co-

¹Notar que todas las restricciones son tratadas como duras, ya que su satisfacción se considera imprescindible

mo solución el punto máximo de la densidad en $\hat{x} = (0.74, 0.5)$, el cual evaluado en las restricciones tiene un valor de $f_{res}(\hat{x}) = 0.047599999999999976$ y evaluado en la función objetivo tiene un valor de $f_{obj}(\hat{x}) = 0.7976$. Después se procede a obtener una población de soluciones tomando el punto máximo de la densidad y un conjunto de puntos muestreados de la misma densidad. Dicha población de puntos es adaptada por el algoritmo genético usando la función objetivo. Después de iterar el algoritmo genético, se encontró como la mejor solución adaptada el punto $x^* = (0.2365010937885068, 0.5)$, el cual es evaluado en la función objetivo y se obtiene un valor de $f_{obj}(x^*) = 0.3059327673631601$ y evaluado en la función de restricciones tiene un valor de $f_{res}(x^*) = 0.567$. Con esto se puede notar que el punto que adaptó el algoritmo genético mejoró en su valor objetivo pero empeoró en el valor objetivo de la función de restricciones. Se tiene la sospecha de que esto ocurre porque la población se expande fuera de la frontera de la densidad, donde también se encuentran buenos valores objetivos locales como se puede observar ese comportamiento en la Figura 4.11 a través de las iteraciones. Por tal motivo es necesario mejorar el operador de cruce para que la población sea contraída en la región de alta probabilidad.

4.9 CONCLUSIONES DEL CAPÍTULO

Para el estudio del SFPL con restricciones, se propusieron diferentes enfoques para el manejo de las restricciones. En primera instancia, se propuso un método de penalización, el cual no resultó ser muy eficiente de forma general para diversas instancias de un problema de la mochila. Dado lo anterior, se optó por buscar otras formas para tratar a las restricciones. Inspirado en una formulación para un problema bicriterio, se propuso dividir el problema de optimización original en dos partes. La primera parte es tratar solo a la función objetivo como tal y la segunda es tratar solo a la función de restricciones. De esta forma se procede a estimar las densidades para cada parte del problema con su respectiva función. Como resultado se obtiene una densidad conjunta proveniente de las densidades anteriores. Esta densidad conjunta

resultante describe perfectamente las regiones donde se encuentra la solución óptima global. Donde los puntos tomados de esa densidad satisfacen a las restricciones con un buen valor en el objetivo.

Para una población de puntos que es tomada de la densidad conjunta se procede a explorar la densidad estimada por el SFPL en la región de máxima probabilidad mediante un algoritmo genético. El propósito de explorar esta región es el de encontrar una solución cuyo valor objetivo sea el mejor, o bien el valor mejor adaptado en esa región. Esta idea se propone de dos formas. La primera es el algoritmo SFPL-AGI cuyo mecanismo consiste en pasar una población inicial a un algoritmo genético tras terminar las iteraciones del SFPL. La segunda forma es el SFPL-AGII que tras cada iteración del algoritmo SFPL se extrae una población de puntos y se pasa a un algoritmo genético. La mejor solución obtenida por el algoritmo genético se guarda en una lista y se procede a la siguiente iteración del SFPL. Al final del algoritmo SFPL-AGII lo que se tiene es una lista de las mejores soluciones en cada iteración del SFPL. Con lo anterior se procede a evaluar todas las soluciones y se toma la solución con el mejor valor objetivo. Por otro lado, esta lista podría servir para el enfoque bicriterio discutido en este capítulo.

Por último, dado el desempeño del SFPL con los algoritmos genéticos se propone el SFPL como un mecanismo para manejar las restricciones para los algoritmos genéticos. En experimentos preliminares se observó que este enfoque tiene un desempeño muy prometedor ya que se obtuvieron buenos resultados.

Con lo anterior, se tiene que el algoritmo SFPL tiene el potencial de mejorar sustancialmente el desempeño de los algoritmos genéticos efectuando la diversificación de la solución en las regiones donde los óptimos globales se encuentran con alta probabilidad.

CAPÍTULO 5

EXPERIMENTACIÓN COMPUTACIONAL

El algoritmo SFPL busca proporcionar una herramienta más para la solución de problemas de optimización global, la cual es la función de densidad de probabilidad asociada al problema tratado que a su vez es adaptada por los algoritmos genéticos.

El algoritmo SFPL se aplica en distintos problemas de optimización no lineal utilizándolo como generador de puntos iniciales para algoritmos genéticos. Esto con el propósito de mostrar que al utilizar puntos generados mediante la función de densidad obtenida por SFPL se presenta una mejora notoria en la calidad de las soluciones obtenidas por los algoritmos genéticos. Esto solo es posible si la densidad obtenida mediante SFPL es acorde con la densidad del problema de optimización tratado.

Una de las ventajas de SFPL es que se puede controlar la complejidad computacional que se desea. El algoritmo requiere de $2(L - 1)MN$ evaluaciones de la función objetivo por lo que se pueden obtener funciones de densidad tan ajustadas como se desee.

Las pruebas consisten en dos principales experimentos. En el primer experimento, se presenta un análisis estadístico de los resultados típicos obtenidos por los algoritmos genéticos y el SFPL híbrido con el fin de evaluar el desempeño de estos enfoques propuestos en previos capítulos en tres problemas muy conocidos para la optimización global. En el segundo experimento, se realiza una comparación del desempeño del SFPL contra los algoritmos genéticos en un problema que crece en

dimensión, esto con el fin de evaluar el comportamiento en tiempo de cómputo con respecto al tamaño del problema.

Todos los experimentos implementados en el presente trabajo se ejecutaron en una computadora personal con procesador Intel Centrino Core 2 Duo a 1.50 Ghz. con memoria DDR2, 1 Gb, 667 Mhz.

5.1 ANÁLISIS ESTADÍSTICO

Se realizó el control de diferentes e independientes condiciones de inicio para el algoritmo SFPL. Esto con el fin de observar que la convergencia al estado estacionario sea la misma, y para medir la variabilidad de los experimentos. Como resultado se obtuvo una densidad que converge a la misma región dentro de la región generada por los parámetros L y D predefinidos. Esto ocurre debido a que en un muestreo Gibbs las densidades, que son independientes a los puntos de inicio, aparecen contraídas en la región del espacio de búsqueda [5].

5.1.1 FUNCIÓN DE BOOTH

El problema de Booth [22] es uno de los más usados para evaluar el desempeño de los algoritmos heurísticos y por tal hecho se usa en el presente trabajo para evaluar el desempeño del SFPL-AGI híbrido y el AGS. En la Figura 5.1 se muestra gráficamente la función de Booth en 2D. La función se describe de la siguiente forma,

Booth:

$$\begin{aligned} \text{mín} \quad & f(x_1, x_2) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2, \\ \text{sujeta a} \quad & -10 \leq x_1 \leq 10, \\ & -10 \leq x_2 \leq 10, \end{aligned}$$

Solución: $x^* = (1, 3)$, $f(x^*) = 0$.

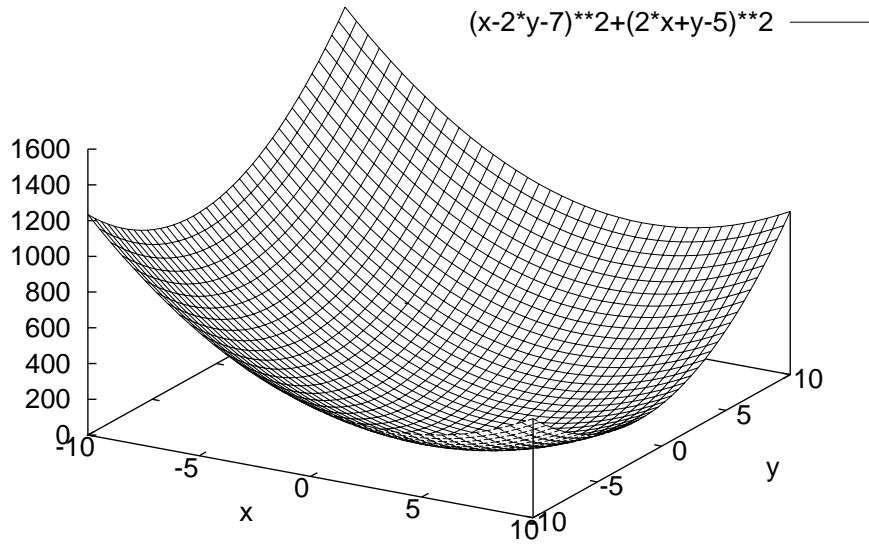


Figura 5.1: Función de Booth en 2D, $f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$.

A continuación se compara el desempeño del algoritmo SFPL-AGI con el AGS en la función de Booth.

Método	Valor obj. It=50	Valor obj. It=100	Valor obj. It=200
AGS	0.0050 , 0.0090	0.0052 , 0.0066	0.0100 , 0.0269
SFPL-AGI	0.0000 , 0.0000	0.0000 , 0.0001	0.0000 , 0.0000

Tabla 5.1: Resultados típicos en 10 corridas de los algoritmos AGS y SFPL-AGI con 50, 100 y 200 iteraciones en cada algoritmo para el problema de Booth.

En la Tabla 5.1 se muestran los resultados típicos obtenidos en 10 corridas para cada algoritmo con 50, 100 y 200 iteraciones. Para las tablas de esta sección, los valores de las tres columnas de la derecha corresponden a los valores objetivos promedios y los valores de la desviación estándar, respectivamente. El algoritmo AGS en todas sus iteraciones encuentra una solución muy aproximada con un buen valor objetivo. Asimismo se observa que la variabilidad de los resultados entre corridas son

muy pequeños. Por lo que se espera que siempre obtenga buenos resultados. Por otro lado, el algoritmo SFPL-AGI encuentra la solución óptima en todas las iteraciones y su variabilidad es nula. Con esto se tiene que el algoritmo SFPL híbrido mejora los resultados del algoritmo genético. Sin embargo el desempeño del algoritmo híbrido en este problema no es muy notorio ya que este problema es relativamente sencillo de resolver con los algoritmos heurísticos. El tiempo de cómputo para el SFPL-AGI es menor al minuto para todas sus iteraciones y el AGS es menor a los cinco segundos para todas sus iteraciones.

5.1.2 FUNCIÓN DE ROSENBROCK

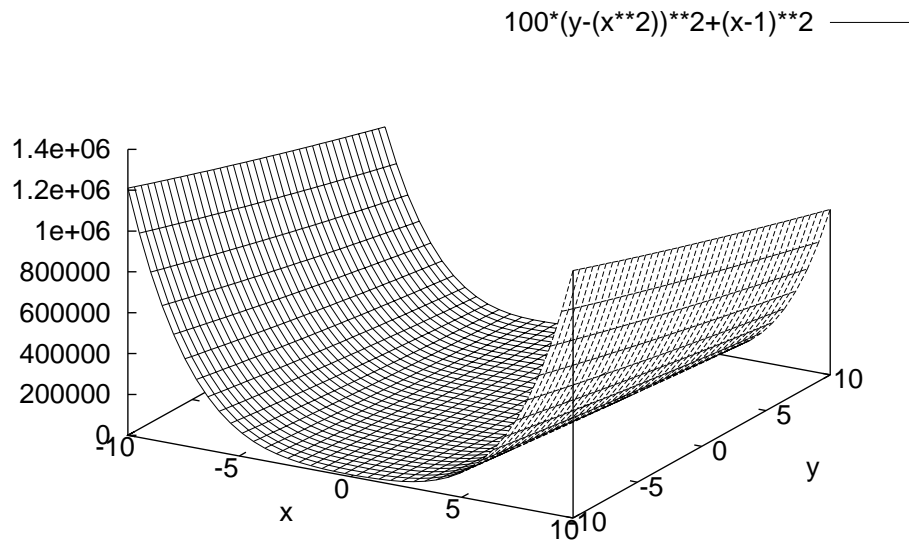


Figura 5.2: Función de Rosenbrock en 2D, $f(x, y) = 100(y - x^2)^2 + (x - 1)^2$.

En la función de Rosenbrock [22] la solución óptima global se localiza dentro de un largo y estrecho valle. Para encontrar el valle es trivial, sin embargo la convergencia hacia una solución óptima global es difícil, por lo tanto, este problema es utilizado frecuentemente para probar el rendimiento de los algoritmos de optimización. En la Figura 5.2 se muestra gráficamente la función de Rosenbrock en 2D.

La función se describe de la siguiente forma,

Función de Rosenbrock:

$N = 20$,

$$\begin{aligned} \min \quad & f(x) = \sum_{n=1}^N 100(x_{n+1} - x_n^2)^2 + (x_n - 1)^2, \\ \text{sujeta a} \quad & -10 \leq x_n \leq 10, \quad n = 1, \dots, N \end{aligned}$$

Solución: $x^* = (1, \dots, 1)$, $f(x^*) = 0$.

Es importante mencionar aquí que el problema Rosenbrock 20-dimensional, es reportado por Laguna [22] como muy difícil de resolver por algoritmos heurísticos como los algoritmos genéticos.

En el presente trabajo se compara el desempeño del algoritmo SFPL híbrido con los algoritmos genéticos en la función de Rosenbrock.

Método	Valor obj. It=50	Valor obj. It=100	Valor obj. It=200
AGS	1330.55 , 492.72	508.32 , 188.33	190.79 , 56.89
SFPL-AGI	30.90 , 6.16	19.89 , 2.47	18.49 , 0.43

Tabla 5.2: Resultados típicos en 10 corridas de los algoritmos AGS y SFPL-AGI con 50, 100 y 200 iteraciones en cada algoritmo para el problema de Rosenbrock.

En la Tabla 5.2 se muestran los resultados típicos obtenidos en 10 corridas para cada algoritmo con 50, 100 y 200 iteraciones. Las celdas muestran la misma información que la Tabla 5.1 (promedio, desviación estándar). El desempeño del algoritmo AGS en todas sus iteraciones encuentra soluciones con valores objetivo muy altos lejanos al óptimo conocido. Asimismo, la variabilidad entre corridas independientes resulta en valores relativamente grandes, lo cual confirma la dificultad mencionada por Laguna para el AGS probado. Por otro lado, el algoritmo SFPL-AGI encuentra soluciones mucho mejores que el AGS, aunque tampoco llega a la solución óptima. La

variabilidad de los resultados entre corridas es relativamente pequeña con respecto a la del AGS. Con esto se observa que el algoritmo SFPL-AGI mejora los resultados del algoritmo AGS. Para ambos problemas la función de Rosenbrock es muy difícil de optimizar dadas las características de esta función. El tiempo de cómputo para el SFPL-AGI es menor a los seis minutos para todas sus iteraciones y el AGS es menor a los diez segundos para todas sus iteraciones.

5.1.3 FUNCIÓN DE SCHWEFEL

La función de Schwefel [22] tiene la característica que sus dos mejores mínimos locales están relativamente distantes el uno del otro. Por lo tanto, los algoritmos de búsqueda son muy propensos a tener una convergencia en la dirección equivocada. En la Figura 5.4 se puede ver que la estimación de las densidades reflejan las dos regiones muy bien distinguidas.

Por otro lado, el problema de Schwefel es un ejemplo de una función separable, es decir, una función donde cada término envuelve una sola variable simple. Dado que la búsqueda estocástica dada por la ecuación (3.1) se realiza para cada variable como un sistema dinámico, solo se requiere de una iteración del SFPL para su convergencia en este tipo de problemas (separables). En la Figura 5.3 se muestra gráficamente la función de Schwefel en 2D. La función se describe de la siguiente forma,

Función de Schwefel:

$$N = 6,$$

$$\begin{aligned} \text{mín} \quad & f(x) = 418.9829N - \sum_{n=1}^N x_n \sin(\sqrt{|x_n|}), \\ \text{sujeta a} \quad & -500 \leq x_n \leq 500, \quad n = 1, \dots, N \end{aligned}$$

$$\text{Solución: } x^* = (420.9687, \dots, 420.9687), f(x^*) = 0.$$

A continuación se compara el desempeño del algoritmo SFPL híbrido con los algoritmos genéticos en la función de Schwefel.

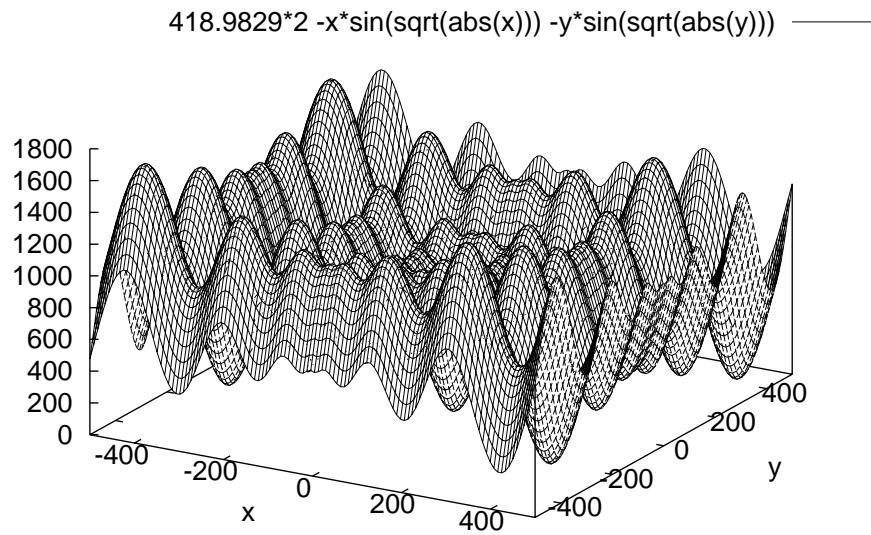


Figura 5.3: Función de Schwefel en 2D, $f(x, y) = -x \sin(\sqrt{|x|}) - y \sin(\sqrt{|y|})$.

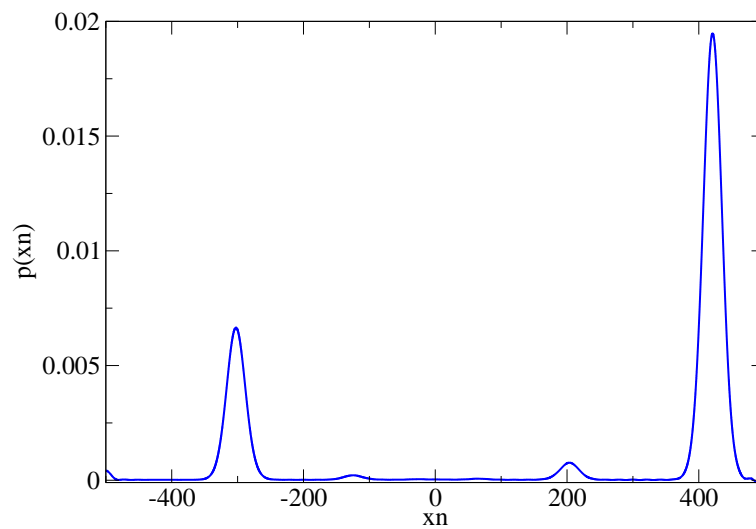


Figura 5.4: Estimación de las densidades para el problema de Schwefel con $L = 100$ y $D = 50$ donde muestra las dos regiones mínimas muy bien distinguidas.

Método	Valor obj. It=50	Valor obj. It=100	Valor obj. It=200
AGS	169.65 , 84.08	21.66 , 13.55	5.47 , 4.53
SFPL-AGI	0.0001 , 0.0000	0.0001 , 0.0000	0.0001 , 0.0000

Tabla 5.3: Resultados típicos en 10 corridas de los algoritmos AGS y SFPL-AGI con 50, 100 y 200 iteraciones en cada algoritmo para el problema de Schwefel.

En la Tabla 5.3 se muestran los resultados típicos obtenidos en 10 corridas para cada algoritmo con 50, 100 y 200 iteraciones. Las celdas muestran la misma información que la Tabla 5.1 (promedio, desviación estándar). El desempeño del algoritmo AGS para 200 iteraciones encuentra valores objetivo pequeños más aproximados. La variabilidad entre corridas para las 200 iteraciones del AGS es relativamente pequeña. A pesar que la función de Schwefel es complicada para algoritmos heurísticos como los genéticos, se pueden encontrar buenas soluciones con ellos. Por otro lado el algoritmo SFPL-AGI encuentra la solución óptima eficientemente ya que solo bastaba con una iteración del SFPL y 14 iteraciones del algoritmo de adaptación para lograrlo. La variabilidad en todas las iteraciones es nula. Con esto se puede ver que el algoritmo SFPL-AGI híbrido mejora los resultados del AGS en este caso particular de una funciones con característica de separabilidad. El tiempo de cómputo para ambos métodos es menor a los dos segundos para cada una de sus réplicas.

5.2 COSTO COMPUTACIONAL Y DIMENSIONALIDAD

5.2.1 ALGORITMOS GENÉTICOS

En la Tabla 5.4 se muestran los resultados del experimento realizado con el algoritmo AGS aplicado al problema de Rosenbrock creciéndolo en dimensión donde las soluciones se conocen. Para este se probaron dimensiones de 20, 40, 60, 80 y 100 variables con 50, 100 y 200 iteraciones del algoritmo genético en el problema mencionado. Se observa que el algoritmo AGS no encuentra buenos valores objetivos de la función. Conforme crece la dimensión los resultados obtenidos para los valores

objetivos son peores, por lo que encontrar una buena solución aproximada es difícil. Esto se debe a que la función de Rosenbrock es muy complicada para el algoritmo AGS. Como anteriormente se mencionó, la función de Rosenbrock de 20 dimensiones es muy complicada de resolver por el AGS, por lo que para dimensiones grandes es aún más complicada.

Dimensión	Valor obj. It=50	Valor obj. It=100	Valor obj. It=200
20	969.54	395.61	196.97
40	32835.27	12357.93	6738.83
60	56690.75	40437.75	30360.77
80	147024.07	112551.91	96982.19
100	283819.65	217443.38	198826.08

Tabla 5.4: Resultados del algoritmo genético para el problema de Rosenbrock con tamaños de dimensión de 20, 40, 60, 80 y 100.

5.2.2 SFPL HÍBRIDO

La Tabla 5.5 muestra los resultados del experimento realizado con el algoritmo SFPL-AGI híbrido aplicado al problema de Rosenbrock con 20, 40, 60, 80 y 100 dimensiones. Los valores objetivos encontrados para la dimensión de 20 variables se aproxima mucho más a la solución con respecto a la solución encontrada por los algoritmos genéticos. Conforme crece la dimensión también se le dificulta al algoritmo SFPL híbrido encontrar buenas soluciones. El algoritmo SFPL híbrido sin duda obtiene mejores valores objetivos para la función de Rosenbrock que los algoritmos genéticos. Esto se debe a que se explora en una región dada por el SFPL donde las soluciones se encuentran en la vecindad del óptimo global del problema.

En las Figuras 5.5, 5.6 y 5.7 se muestra el comportamiento en tiempo de cómputo con respecto a la dimensión del Rosenbrock. Se tienen tamaños de dimensión de 20, 40, 60, 80 y 100 con número de iteraciones de 50, 100 y 200. Se observa que

el tiempo de cómputo de la implementación del algoritmo SFPL-AGI no tiene un crecimiento exponencial con respecto a la dimensión del problema para los rangos de operación definidos. En las Figuras 5.6 y 5.7 se puede notar que el tiempo de cómputo de la dimensión 100 decae con respecto al tiempo de la iteración 80. Se desconoce el comportamiento descrito ya que para tener una conclusión consistente es necesario realizar más réplicas del algoritmo SFPL-AGI.

Dimensión	Valor obj. It=50	Valor obj. It=100	Valor obj. It=200
20	25.00	18.29	17.29
40	140.46	59.06	44.45
60	390.92	143.95	96.65
80	650.47	175.56	192.26
100	980.12	556.73	486.60

Tabla 5.5: Resultados del algoritmo SFPL híbrido para el problema de Rosenbrock con tamaños de dimensión de 20, 40, 60, 80 y 100.

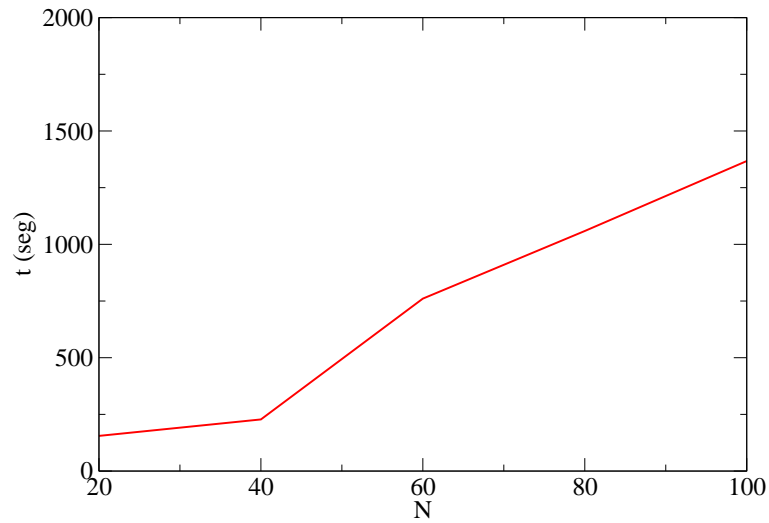


Figura 5.5: Comportamiento del tiempo de cómputo para el algoritmo SFPL híbrido en 50 iteraciones con respecto a la dimensión del problema de Rosenbrock.

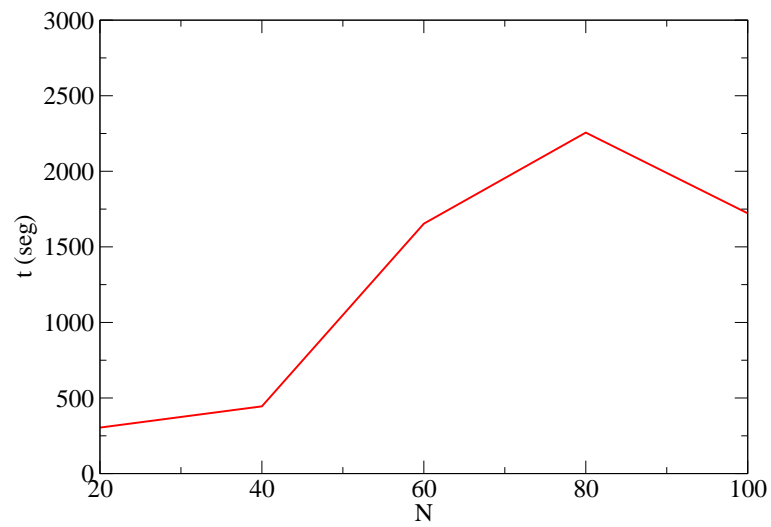


Figura 5.6: Comportamiento del tiempo de cómputo para el algoritmo SFPL híbrido en 100 iteraciones con respecto a la dimensión del problema de Rosenbrock.

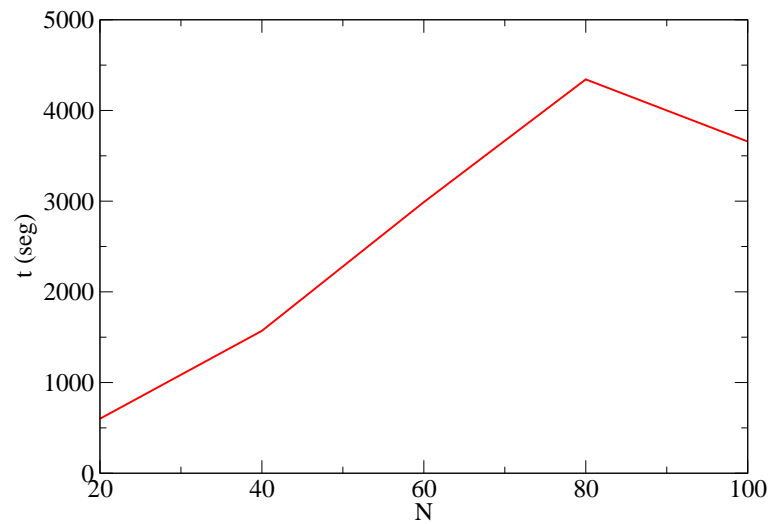


Figura 5.7: Comportamiento del tiempo de cómputo para el algoritmo SFPL híbrido en 200 iteraciones con respecto a la dimensión del problema de Rosenbrock.

5.3 CONCLUSIONES DEL CAPÍTULO

El potencial del algoritmo SFPL refleja las regiones donde se contiene a la solución óptima global con alta probabilidad. Esto se puede observar ya que obteniendo una población de puntos de esa región es posible encontrar una solución tras adaptar esos puntos con algún mecanismo adaptativo como los algoritmos genéticos. Si se procede de esta forma ya no es necesario explorar todo el espacio de búsqueda ya que el algoritmo de estimación ya lo observó, y definió las regiones preferentes donde existen las soluciones óptimas locales. La solución óptima global esta definida en la región con máxima probabilidad. La región de máxima probabilidad es la región de interés y es donde se enfocan los algoritmos propuestos. Esto es sin duda una herramienta más para algoritmos de optimización global. Los resultados de la experimentación muestran que estas regiones sí son de gran ayuda para que los algoritmos genéticos obtengan soluciones de forma eficiente y de calidad. Para la función de Rosenbrock no se logró encontrar una buena solución aproximada. Esto se debe a que el mecanismo de adaptación utilizado podría no ser el mejor para el SFPL, por lo que probar otros mecanismos adaptativos sería muy conveniente. También se podría tener un algoritmo genético más ajustado en el sentido que la parte adaptativa del algoritmo genético no se expanda en población. Esto es lo que requiere para que la búsqueda solo se realice en la región definida por el algoritmo SFPL.

Por otro lado, también se observó que el algoritmo SFPL híbrido requiere de un tiempo de cómputo con respecto a la dimensión del problema bien comportado en el rango de operación, ya que a los resultados obtenidos se les ajustó un polinomio de orden 3 y 4. Empíricamente se determinó que el tiempo de cómputo del algoritmo SFPL no es consistente con un crecimiento exponencial en los tamaños de instancia probados. Para llegar a una conclusión consistente del tiempo de cómputo del algoritmo SFPL-híbrido, con la muestra observada no es suficiente. Para esto se requiere realizar más experimentos.

CAPÍTULO 6

CONCLUSIONES

6.1 CONTRIBUCIONES

Se ha propuesto una nueva metodología para tratar a los problemas de optimización global a gran escala con restricciones. Se propone el algoritmo de estimación de densidad como un mecanismo para brindar poblaciones de puntos iniciales a los algoritmos heurísticos. Se construye un algoritmo híbrido, a partir del algoritmo de estimación de densidad y los algoritmos genéticos, que brinda soluciones de buena calidad en un tiempo de cómputo razonable. Se propone un nuevo enfoque del algoritmo de estimación de densidad como mecanismo para el manejo de las restricciones en los algoritmos genéticos.

Los principales resultados de esta tesis han sido publicados en: Learning probability densities of optimization problems with constraints and uncertainty. En Advances in Artificial Intelligence, Lecture Notes in Computer Science, Springer. Y presentado en el 7th. Mexican International Conference on Artificial Intelligence 2008 (MICA), en Atizapán de Zaragoza del 27 al 31 de Octubre. Asimismo fueron presentado en el XLV Congreso Nacional de la Sociedad Matemática Mexicana, en Valle de Bravo Cd. de México, del 20 al 24 de Octubre de 2008, y en los seminarios de investigación del programa de Posgrado en Ingeniería de Sistemas el 8 de Mayo y 16 de Octubre de 2008.

6.2 CONCLUSIONES

Se logró construir una herramienta para tratar a los problemas de optimización global con y sin restricciones a través de un algoritmo híbrido que brinda soluciones de buena calidad en tiempo de cómputo razonable para los casos probados. El algoritmo de estimación de densidad encuentra, en estos casos de prueba, las regiones que con alta probabilidad contienen el óptimo global. Dichas regiones definidas se utilizan para ser exploradas por un algoritmo genético. Con esto se tiene que el algoritmo de estimación de densidad mejora sustancialmente el desempeño de los algoritmos genéticos ya que no es necesario realizar la búsqueda en todo el dominio, sino en las regiones de máxima probabilidad. Por tal razón, el algoritmo híbrido brinda soluciones que son consistentes con las propiedades óptimas globales. El desempeño del algoritmo híbrido se ha comparado en primera instancia con un algoritmo genético para la optimización global, por la razón de querer observar el desempeño de los algoritmos genéticos con una población inicial aleatoria y una población inicial dada por el algoritmo de estimación de densidad en regiones de máxima probabilidad. Con esto se tiene que dada una solución inicial en la región de máxima probabilidad el algoritmo genético converge eficientemente a una solución de buena calidad. El método propuesto puede servir para la optimización global por el hecho de que estima densidades al estado estacionario, lo que sería equivalente a realizar una búsqueda en un tiempo infinito. Con esto se tiene un panorama global de máximos y mínimos del espacio de búsqueda. Dado lo anterior, se abre puerta a el desarrollo de nuevos algoritmos o la mejora de algoritmos deterministas y estocásticos existentes.

6.3 TRABAJO FUTURO

Dados los resultados obtenidos previamente del desempeño del algoritmo propuesto, se tiene que es una herramienta muy prometedora para la optimización global que valdría la pena seguir explorándola. Por lo que se dejan los siguientes puntos

como áreas de oportunidad para trabajo futuro en esta línea de investigación.

- Probar una mayor variedad de funciones de prueba para la optimización global con y sin restricciones.
- Probar el desempeño del SFPL con otros algoritmos heurísticos y exactos para la optimización global. Por ejemplo algunos algoritmos heurísticos que han reportado buenos resultados en problemas de este tipo son búsqueda tabu (*Tabu Search*), búsqueda local ascendente (*Hill Climbing*), búsquedas en espacios de estados (*State Space Search*), entre otros. En términos de métodos exactos para optimización global hoy en día se cuenta con técnicas muy eficientes como BARON [9] desarrollado por Sahinidis y Tawarmalani y LGO [9] desarrollado por Pintér, con las cuales merece la pena la comparación..
- Aplicar el algoritmo propuesto en casos de estudio no triviales y de alta dimensionalidad. Se ha visto que la verdadera efectividad de un procedimiento heurístico sólo puede comprobarse al estudiar problemas de alta dimensionalidad.
- Desarrollar un enfoque más apropiado para el manejo de restricciones en algoritmos heurísticos evolutivos. Por ejemplo, los algoritmos evolutivos actuales hoy en día trabajan con una población muy grande de individuos y presentan en ocasiones dificultades en problemas con restricciones. Estos algoritmos podrían beneficiarse de técnicas como la propuesta para ser más selectivos en la elección de los individuos tratando de sesgar sus probabilidades de generación hacia individuos que satisfagan con alta probabilidad a las restricciones.
- Explorar diferentes mecanismos para la adaptación de las soluciones del algoritmo de estimación de densidad. Dado que solo se exploró un mecanismo de adaptación, es de bastante interés encontrar el más adecuado y eficiente para el SFPL.

-
- Eficientar el algoritmo mediante el uso de estructuras de datos internas más apropiadas.
 - Paralelizar el algoritmo de estimación de densidad. Esto puede ser muy ventajoso ya que varios de las operaciones del método pueden efectuarse de forma independiente.

APÉNDICE A

MÉTODO DEL GRADIENTE

Para tener claro el método del gradiente, necesitamos recordar algunos conceptos básicos que a continuación son definidos [21].

DERIVADA PARCIAL

Dada una función escalar $f(x, y)$, definimos una derivada parcial de f respecto a la variable x , $\frac{\partial f}{\partial x}$, como la derivada de f respecto a x , suponiendo que la variable y es constante.

La pendiente en la dirección de y esta dada por la derivada parcial de $f_y(x, y)$, y la pendiente en la dirección de x esta dada por la derivada parcial de $f_x(x, y)$. Entonces con dos derivadas parciales se pueden usar para calcular la pendiente en cualquier dirección.

DERIVADA DIRECCIONAL

Si f es una función diferenciable en el punto de x y y , entonces la derivada direccional de f en la dirección del vector unitario u que se denota por D_u , al siguiente limite (si existe) se define por,

$$D_u f(x, y) = \lim_{h \rightarrow 0} \frac{f(x + hu, y + hu) - f(x, y)}{h}.$$

Para funciones de dos variables, un vector unitario se puede representar de la

forma $[\cos \theta, \sin \theta]$ y se habla de la derivada direccional al correspondiente ángulo θ . Se escribe como $u = \cos \theta i + \sin \theta j$, donde i y j son los vectores unitarios $(1, 0)$ y $(0, 1)$, respectivamente.

La derivada direccional se puede expresar en función de las derivadas parciales como,

$$D_u f(x, y) = f_x(x, y) \cos \theta + f_y(x, y) \sin \theta.$$

La derivada direccional también se puede interpretar como la pendiente en la superficie, en la dirección del vector unitario u .

EL GRADIENTE

Sea $f(x, y)$ una función de x y y tal que f_x y f_y existen. Entonces el gradiente de f denotado por $\nabla f(x, y)$ es el vector,

$$\nabla f(x, y) = f_x(x, y)i + f_y(x, y)j.$$

Como el gradiente de f es un vector, se puede expresar la derivada direccional de f en la dirección de u como,

$$D_u f(x, y) = \{f_x(x, y)i + f_y(x, y)j\} \cdot \{\cos \theta i + \sin \theta j\}.$$

En otras palabras, la derivada direccional es el producto escalar del gradiente y el vector dirección.

Teorema A.1 *Si f es una función diferenciable de x y y , entonces la derivada direccional de f en la dirección del vector unitario u es,*

$$D_u f(x, y) = \nabla f(x, y) \cdot u.$$

PROPIEDADES DEL GRADIENTE

- $\nabla f(x, y) = 0$, entonces $D_u f(x, y) = 0$ para toda u .
- La derivada direccional en (x, y) es máxima en la dirección del vector gradiente $\nabla f(x, y)$ (dirección de máximo incremento de f), siendo $\|\nabla f(x, y)\|$ su valor máximo.
- La derivada direccional en (x, y) es mínima en la dirección del vector $-\nabla f(x, y)$ (dirección de mínimo incremento de f), siendo $-\|\nabla f(x, y)\|$ su valor mínimo.
- La derivada direccional en (x, y) es nula en cualquier dirección perpendicular al vector gradiente.

El gradiente es la generalización de una derivada, y la dirección está dada por la combinación lineal de ambos vectores como se muestra en la Figura A.1.

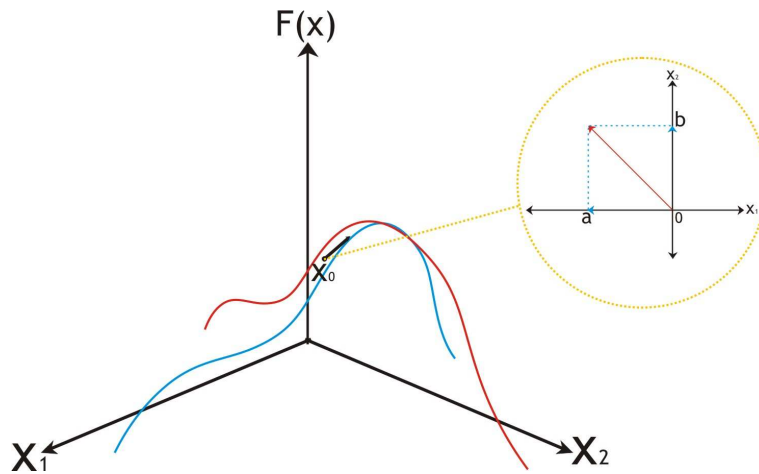


Figura A.1: Representación de una combinación lineal de vectores en el método del gradiente.

APÉNDICE B

RECOCIDO SIMULADO

El método del recocido simulado propuesto por Kirkpatrick et al. [19], es utilizado en la industria para obtener materiales mas resistentes, o mas cristalinos, en general, para mejorar las cualidades de un material. El proceso consiste en derretir el material calentándolo a muy alta temperatura. En esa situación, los átomos adquieren una distribución azarosa dentro de la estructura del material y la energía del sistema es máxima. Luego se hace descender la temperatura muy lentamente por etapas, dejando que en cada una de esas etapas los átomos queden en equilibrio. Es decir, que los átomos alcancen una configuración óptima para esa temperatura. Al final del proceso, los átomos forman una estructura cristalina altamente regular, el material alcanza así una máxima resistencia y la energía del sistema es mínima.

Experimentalmente se comprueba que si la temperatura se hace descender bruscamente, o bien, si no se espera suficiente tiempo en cada etapa, al final la estructura del material no es la óptima. La rama de la física llamada Mecánica Estadística, se encargo de desarrollar una serie de métodos para estudiar el comportamiento de grandes cantidades de átomos de un sistema. Debido a que en promedio, en un sistema hay 10^{23} átomos por centímetro cúbico, solamente puede estudiarse el comportamiento mas probable del sistema en equilibrio a una dada temperatura. La experimentación mostró que los átomos de un sistema en un proceso de recocido se comportan según el factor de probabilidad de Boltzman. En días muy tempranos de la computación científica, Metrópolis modelo el proceso de recocido simulado, donde en cada paso del algoritmo se le da al átomo un pequeño desplazamiento azaroso y

se mide el cambio de energía ΔE . Si $\Delta E \leq 0$ se acepta el desplazamiento y la configuración con ese desplazamiento es usado como punto de inicio del siguiente paso. Si $\Delta E > 0$, se acepta el desplazamiento con probabilidad $P(\delta E) = \exp(-\Delta E/T \cdot K)$, donde T es la temperatura del sistema y K es la constante de Boltzman [33, 37]. Al principio T es un valor alto (en fundición) y luego va disminuyendo (en enfriamiento), es decir, cada vez es menor la probabilidad de que la nueva alternativa reemplace a la anterior.

APÉNDICE C

LISTA DE TÉRMINOS Y DEFINICIONES

ESPACIO DE BÚSQUEDA: Este es el conjunto de todas las soluciones factibles para un problema de optimización.

HEURÍSTICA: Se refiere a un algoritmo que encuentra soluciones de buena calidad a un problema en un tiempo de cómputo razonable.

EQUILIBRIO TÉRMICO: Estado en el cual un sistema físico está descrito por las medidas de probabilidad que son independientes con el tiempo.

PROCESO POR DIFUSIÓN: Desplazamiento aleatorio de las partículas en un sistema físico debido a la acción de una temperatura.

ALGORITMOS DE BÚSQUEDA ESTOCÁSTICA: Se refiere a los algoritmos de optimización que incorporan aleatoriedad en la exploración de los espacios de búsqueda.

DIVERSIFICACIÓN: Procedimiento para generar diferentes poblaciones de soluciones en la región de búsqueda.

ADAPTACIÓN: Procedimiento para llevar o contraer a las soluciones a espacios de la región de búsqueda donde tomen su mejor valor.

BIBLIOGRAFÍA

- [1] Bagnoli F., Berrones A., Franci F.: Degustibus disputandum (forecasting opinions by knowledge networks). *Journal of Physica*, 332:509-518, 2004.
- [2] Berrones A.: Generating random deviates consistent with the long term behavior of stochastic search processes in global optimization. En: *Proceedings of the 9th International Work-Conference on Artificial Neural Networks (IWANN 2007)*, Lecture Notes in Computer Science, Vol. 4507, pp. 1-8, Springer, Berlín, Alemania, 2007.
- [3] Berrones A.: Stationary probability density of stochastic search processes in global optimization. *Journal of Statistical Mechanics*, P01013, 2008.
- [4] Birge J. R., Louveaux F.: *Introduction to Stochastic Programming*. Springer-Verlang, Berlín, Alemania, 1997.
- [5] Canty A.: Hypothesis tests of convergence in markov chain monte carlo. *Journal of Computational and Graphical Statistics*, 8:93-108, 1999.
- [6] Darwin C.: *On the Origin of Species by Means of Natural Selection*. Murray Edit, Londres, Inglaterra, 1859.
- [7] De Jong K. A.: *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems*. Tesis doctoral, University of Michigan, Ann Arbor, EUA, 1975.
- [8] Floudas C. A.: *Deterministic Global Optimization: Theory, Methods and Applications*. Kluwer, Boston, EUA, 2000.

-
- [9] GAMS Development Corporation.: *GAMS: The Solver Manuals*. Washington, EUA, 2008.
 - [10] Geman S., Geman D.: Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721-741, 1984.
 - [11] Geman S., Hwang C. R.: Diffusions for global optimization. *SIAM Journal of Control Optimization*, 24(5):1031-1043, 1986.
 - [12] GNU Grace, <http://plasma-gate.weizmann.ac.il/Grace/doc/>
 - [13] GNU Linear Programming Kit, <http://www.gnu.org/software/glpk/>
 - [14] GNU Plot, <http://www.gnuplot.info/>
 - [15] Goldberg D.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Boston, EUA, 1989.
 - [16] Grasman J., Van Herwaarden O. A.: *Asymptotic Methods for the Fokker-Planck Equation and the Exit Problem in Applications*. Springer-Verlag, Berlín, Alemania, 1999.
 - [17] Holland J.: *Adaptation in Natural and Artificial Systems*. Tesis doctoral, University of Michigan, Ann Arbor, EUA, 1975.
 - [18] Kellerer H., Pferschy U., Pisinger D.: *Knapsack Problems*. Springer-Verlag, Berlín, Alemania, 2004.
 - [19] Kirkpatrick S., Gelatt Jr., C. D., Vecchi M. P.: Optimization by simulated annealing. *Science*, 220:671-680, 1983.
 - [20] Kosmatopoulos E. B., Christodoulou M. A.: The Boltzmann g-RHONN: A learning machine for estimating unknown probability distributions. *Journal of Neural Networks*, 7(2):271-278, 1994.

-
- [21] Kreyszig E.: *Matemáticas Avanzadas para Ingeniería*. Limusa Wiley, México, DF, 2006.
- [22] Laguna M., Martí R.: Experimental testing of advanced scatter search designs for global optimization of multimodal functions. *Journal of Global Optimization*, 33(2):235-255, 1995.
- [23] Metropolis N., Rosenbluth A., Rosenbluth M., Teller A., Teller E.: Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087-1092, 1953.
- [24] Michalewicz Z.: A survey of constraint handling techniques in evolutionary computation methods. En: *Proceedings of the 4th Annual Conference on Evolutionary Programming*, pp. 135-155, MIT Press, Cambridge, EUA, 1995.
- [25] Michalewicz Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlín, Alemania, 1996.
- [26] Movellan J. R., McClelland J. L.: Learning continuous probability distributions with symmetric diffusion networks. *Cognitive Science*, 17:463-496, 1993.
- [27] Osyczka A., Krenich S.: Evolutionary algorithms for global optimization. En: Pintér J. (editor). *Global Optimization, Scientific and Engineering Case Studies Series: Nonconvex Optimization and Its Applications*, Vol. 85, Springer, New York, EUA, 2006.
- [28] Papadimitriou C. H.: *Computational Complexity*. Addison-Wesley, Reading, EUA, 1994.
- [29] Pardalos P. M., Schoen F.: Recent advances and trends in global optimization: deterministic and stochastic methods. En: *Proceedings of the Sixth International Conference on Foundations of Computer-Aided Process Design*, DSI 1-2004, pp. 119-131, New Jersey, EUA, 2004.

-
- [30] Parpas P., Rustem B., Pistikopoulos E. N.: Linearly constrained global optimization and stochastic differential equations. *Journal of Global Optimization*, 36(2):191-217, 2006.
- [31] Peña D., Sánchez R., Berrones A.: Stationary Fokker–Planck learning for the optimization of parameters in nonlinear models. En: *Proceedings of the 6th Mexican International Conference on Artificial Intelligence (MICA I 2007)*, Lecture Notes in Computer Science, Vol. 4827, pp. 94-104, Springer, Berlín, Alemania, 2007.
- [32] Pisinger D.: Where are the hard knapsack problems? *Computers and Operations Research*, 32(9):2271-2284, 2005.
- [33] Risken H.: *The Fokker–Planck Equation*. Springer, Berlín, Alemania, 1984.
- [34] Spall C. J.: *Introduction to Stochastic Search and Optimization. Estimation, Simulation and Control*. Wiley-Interscience, New Jersey, EUA, 2003.
- [35] Suykens J. A. K., Verrelst H., Vandewalle J.: On-line learning Fokker–Planck machine. *Journal of Neural Processing Letters*, 7(2):81-89, 1998.
- [36] Toscano Pulido G., Coello Coello A. C.: A constraint-handling mechanism for particle swarm optimization. En: *Proceedings of the 2004 Congress on Evolutionary Computation*, Vol. 2, pp. 1396–1403, Springer, Berlín, Alemania, 2004.
- [37] Van Kampen N. G.: *Stochastic Processes in Physics and Chemistry*. North-Holland, Amsterdam, Holanda, 1992.

FICHA AUTOBIOGRÁFICA

Jonás Velasco Álvarez

Candidato para el grado de Maestro en Ciencias
en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

UNA HEURÍSTICA BASADA EN BÚSQUEDA
ESTOCÁSTICA PARA PROBLEMAS DE
OPTIMIZACIÓN GLOBAL

Nació en Los Mochis, Sinaloa, el 24 de Abril de 1984. Es hijo del Sr. René Velasco Flores y de la Sra. Rosa Martha Álvarez López. Es el más pequeño de la familia que con gran energía y entusiasmo ha salido adelante con la ayuda de las personas que le quieren. Es graduado de la Universidad de Occidente «UdeO», campus Los Mochis, de la carrera Ingeniería Industrial y de Sistemas (2002 – 2006). Estuvo seis meses como practicante en el ingenio azucarero de la ciudad en el área de almacén. Antes de concluir sus estudios de licenciatura, realizó un estancia de investigación en el programa de Posgrado de Ingeniería de Sistemas «PISIS», en la Facultad de Ingeniería Mecánica Eléctrica «FIME», de la Universidad Autónoma de Nuevo León «UANL», bajo la supervisión del Dr. Arturo Berrones, mismo con el que colabora

actualmente, quien también supervisó su tesis de maestría, cursada en el mismo programa. Actualmente es miembro de la Sociedad Mexicana de Inteligencia Artificial «SMIA» a partir de octubre del 2008 y colabora como asistente de investigación en el Centro de Innovación, Investigación y Desarrollo en Ingeniería y Tecnología «CIIDIT» de la UANL.